



En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Université Toulouse III - Paul Sabatier*
Discipline ou spécialité : *Informatique*

Présentée et soutenue par *Florence Pontico*
Le *vendredi 11 juillet 2008*

Titre : *Une méthode de conception basée sur des patrons d'interface pour l'e-Gouvernement*

JURY

Regina Bernhaupt, Professeur Assistant de l'Université de Salzbourg (Autriche)
Joëlle Coutaz, Professeur de l'Université Joseph Fourier, Grenoble (France)
Christelle Farenc, Maître de Conférences de l'IUT Paul Sabatier, Tarbes (France)
Philippe Palanque, Professeur de l'Université Paul Sabatier, Toulouse (France)
Dominique Scapin, Directeur de recherche INRIA, Rocquencourt (France)
Marco Winckler, Maître de Conférences de l'IUT Paul Sabatier, Tarbes (France)

École doctorale : *MITT (Mathématiques, informatique, télécommunications de Toulouse)*
Unité de recherche : *IRIT (Institut de Recherche en Informatique de Toulouse)*
Directeur(s) de Thèse : *Regina Bernhaupt, Philippe Palanque, Marco Winckler*
Rapporteurs : *Jean Vanderdonckt, Professeur de l'UCL, Louvain-la-Neuve (Belgique)*
Dominique Scapin, Directeur de recherche INRIA, Rocquencourt (France)

Remerciements

Les premiers remerciements de cette thèse vont à Marco Winckler, qui a mené mes errances scientifiques avec beaucoup de rigueur et de professionnalisme. Ses conseils ont fait faire du trapèze à mes réflexions, ont conduit à des discussions agitées et stimulantes et j'ai beaucoup appris à son contact. J'ai vécu ma crise d'adolescence scientifique face à lui ; je l'ai franchie grâce à lui. *Ein großes danke*[1] to Regina Bernhaupt who conducted the evaluation part of this work. With great teaching skills, she pushed me on the brink of abyssal questions such as "Imagine I'm your grand mother and tell me… What is your thesis about?"!!! This helped me quite a lot to tidy my mind, I am really grateful for that.

Je remercie également Philippe Palanque pour le chaleureux accueil qu'il réserve aux membres de son équipe et ses conseils nets et efficaces, qu'il s'agisse de recherche, d'organisation de conférences ou de questions quotidiennes (p.ex. lui seul a pu trancher sur l'heure adéquate pour le pot de thèse…). Je pense aussi à Christelle Farenc qui a encadré mon travail de DEA puis les débuts de ma thèse. Son écoute attentive et objective m'a aidée à faire des choix professionnels en connaissance de cause, qu'elle en soit sincèrement remerciée. Merci également à Rémi Bastide, pour son ouverture d'esprit et sa curiosité qui lui offrent un imbattable recul sur les questions scientifiques.

Je tiens à remercier Dominique Scapin et Jean Vanderdonckt d'avoir rédigé, à propos de mon mémoire, des rapports exigeants et encourageants, ponctués de conseils aiguisés qui m'ont aidée à mieux définir ma contribution. Je remercie également Joëlle Coutaz, Christelle Farenc et Quentin Limbourg qui me font le plaisir de participer à mon jury.

Ces travaux ont pu voir le jour grâce à l'allocation MENRT qui m'a été attribuée, puis à un poste d'ATER à l'Université Paul Sabatier. Je tiens à remercier les personnes qui m'ont aidée à obtenir ces conditions de travail idéales, notamment Françoise Adreit, François Barrère et Thierry Montaut qui ont recommandé ma candidature d'ATER. Ma thèse a également fait ses premiers pas dans le cadre du projet WebAUDIT : je remercie Genigraph, et en particulier Olivier Nicolas et Jean-Paul Perez, pour leur curiosité et leur intérêt envers nos travaux académiques.

Cette thèse doit beaucoup à Quentin Limbourg qui m'a accueillie en octobre 2006 au sein de SmalS. J'ai partagé ses activités quotidiennes, menées à la fois avec pragmatisme et exigence scientifique et mon appétit de recherche appliquée s'est ouvert. Cette expérience a eu une importance capitale pour moi, je le remercie très sincèrement de m'avoir donné l'opportunité de la mener. Cerise sur le gâteau, son accueil était excellent, ainsi que celui de Jean-Pierre Norguet, à qui je souhaite une deuxième vie épanouissante. Cette mission scientifique a été possible grâce au soutien de l'action de coopération scientifique et technique MAUSE COST Action 294 : j'en remercie ses membres très sincèrement.

Chez SmalS, en avril 2008, Garifalia Kanellopoulos, Gunther de Neeve et Nicolas Vermeulen m'ont à leur tour accueillie, confortant encore la belle image que j'ai de Bruxelles et des Bruxellois. Merci à eux de m'avoir fait confiance et de m'avoir permis d'offrir un début de validation à mes travaux de thèse : c'était une grande chance et une expérience très enthousiasmante. Merci évidemment à Sélim Dahmane, Pierre Herremans, Marc Lainez, Maxime Lamquet, Juan Ruiz Panadero, Karine Picart, Luc Vanderberken et Karlien Von Ongeval qui ont joué le jeu de l'évaluation, avec beaucoup de sérieux et d'application, dans une ambiance souriante et très sympathique.

Parlant de sympathie, je n'oublie pas les membres ~~du LIIHS~~ de IHCS, passés et présents, qui ont accompagné 5 années de ma vie, et pas des moindres, de DEAtte à aujourd'hui. Un merci particulièrement nourri, sincère et amical à Joseph, le fraternel binôme, l'oreille attentive, la

[1] Un grand merci (selon Google traduction !!)

terreur d'Eclipse, le Berserk à lunettes (NDA : « Le berserk est un guerrier-fauve qui entre dans une fureur sacrée, se rendant capable des plus invraisemblables exploits[1] ». La berserkerie est en particulier observée en phase de rédaction de thèse). Il m'a immensément aidée dans la conception de l'outil, prenant du temps sur la rédaction de sa propre thèse pour faire avancer la mienne : mille mercis, chef !! Merci à Amélie qui m'a conseillée dans mes premiers pas d'enseignante, mais aussi montré le chemin de la terrasse, lieu de pauses particulièrement propice aux discussions à cœur ouvert. Elle m'a mis sur la route d'Anne, Karine, Nissou et Sylvie, en qui j'ai trouvé des amies, je lui en suis reconnaissante. Je pense aussi à mes collocs de bureau, Sandra, Syrine et Christophe pour leurs grains de folie gros comme des pastèques, leurs rires bons pour la santé, leurs cours d'Anglais et d'Arabe peu académiques, et leurs projets fous mais réalisables (le tout-en-un® entre autres objets indispensables). Merci aussi à David, Éric, Guillaume, Jeff, Manu et Xavier : je garde de très bons souvenirs des voyages en conférences, des repas d'équipe, et de l'organisation d'IHM'05 et de TAMODIA'07.

Merci aussi aux étudiants, aux collègues enseignants, aux moniteurs de l'UPS et à ceux rencontrés dans les vivifiantes formations du CIES. Merci au personnel de l'IRIT et de l'école doctorale, Françoise Agar, Evelyne Bony, Véronique Debats, Martine Labruyère, Jean-Pierre Baritaud, et Jean-Claude Debelle.

Je remercie les Zanni pour leur amitié, leur tendresse et leur fantaisie. Je suis fière de les compter parmi mes amis : Cath, Florent, Gilbert, Laure, Laurent, Marie, Philippe, Raphaël, Thierry et Véro. Je n'oublie pas Charlotte Guron-Zonbrowski qui a veillé sur ma santé mentale lors de la dernière semaine de rédaction…

Drine, Isa, Justin et Sylvain, mes quatre piliers, je vous aime aussi fort que vous êtes différents. Merci pour tout. Merci à Drine pour sa présence rassurante et son amitié pétillante.

Aucun mot ne décrit l'infini amour et la reconnaissance que j'ai pour ma famille : ma Maman, mon Papa, ma sœur Maryline et mon beauf/coach Patrice, mon frère Jean-Luc, mes neveux Mathilde et Corentin et ma tante Rosy. Leur amour, leur chaleur, et leur inconditionnel soutien me portent comme sur un nuage et je veux leur dire ici à quel point je les en remercie. Je pense aussi à l'amour que m'ont donné mes très chers grands-parents Marie, Marie et Justin, et à mon grand-père Joseph, disparu avant que je n'apparaisse.

Et merci à Stéphane, de tout mon cœur, de me rendre si heureuse.

[1] http://fr.wikipedia.org/wiki/Berserker (dernier accès : 25 sept. 08)

Table des matières

1. Introduction

L’e-Gouvernement (abréviation de « gouvernement électronique ») s’appuie sur les technologies informatiques pour proposer des relations médiatisées par ordinateurs entre administrations et administrés ; les objectifs annoncés sont : diminution des coûts, meilleure circulation des informations, amélioration des conditions d’archivage, personnalisation des services. L’informatisation de l’administration publique par le biais des applications d'e-Gouvernement sert aussi à améliorer l’image des institutions en corrigeant la réputation d’une bureaucratie lourde et opaque pour ses utilisateurs.

Le champ d’application de l’e-Gouvernement est très vaste et, virtuellement, tous les secteurs de l’administration publique peuvent être concernés. Les avantages de l’e-Gouvernement sont aujourd’hui reconnus dans le monde entier, et sont exploités par le biais d’applications proposant des fonctionnalités très différentes telles que :

- Diffusion d’informations et de formulaires via une application Web (p.ex. mise à disposition de formulaires administratifs en Autriche[1]) ;
- Dématérialisation de procédures administratives via le Web (p.ex. déclaration de terrains cultivables en Inde[2]);
- Constitution de bases électroniques de législations, centralisation de dossiers judiciaires (p.ex. projet Phenix d’informatisation des tribunaux en Belgique (Vuyst et Fairchild 2006)) ;
- Dématérialisation de procédures de vote sur le Web (p.ex. élections municipales d’octobre 2005 en Estonie) ;
- Plateformes de discussions avec des institutions ou d’autres citoyens (p.ex. forum réunissant les victimes d’inondations au Royaume-Uni[3]) ;
- Applications Web permettant le règlement d’une somme due à une institution (p.ex. paiement d’une amende contractée par le biais d’un radar automatisé en France[4]).

Les applications d’e-Gouvernement posent des contraintes bien particulières qui se déclinent en fonction de risques associés (p.ex. niveau de sécurité requis), de l’impact sur le public ciblé (p.ex. acceptabilité par ses utilisateurs), des technologies utilisées (p.ex. Web), etc. Certaines de ces contraintes sont soumises à des réglementations légales pour le respect de la vie privée et le contrôle des informations personnelles diffusées (CNIL 2004) ou encore l’accessibilité à l’application quelque soit le dispositif utilisé (Journal Officiel du 10 avril 2002).

Le terme e-Gouvernement désigne un très vaste domaine d’étude. Le travail de thèse présenté dans ce mémoire est focalisé sur un sous domaine de l’e-Gouvernement : l’e-Procuration. Les applications d’e-Procuration sont destinées à supporter la réalisation en ligne de toute (ou d’une partie de) une procédure administrative. Plus particulièrement, nous nous intéressons ici uniquement au processus de conception des interfaces utilisateurs des applications d’e-Procuration, laissant de côté d’autres problématiques telles que la modélisation de données (Kavadias et Efthimos Tambouris 2003), ou les contraintes liées à la spécification des procédures (Brain, Seltsikas, et Tailor 2005). Cependant, les apports de l’étude réalisée ne sont pas réduits au sous domaine de l’e-Procuration.

Les utilisateurs visés par les applications d’e-Procuration sont essentiellement des administrés et des agents administratifs. Du côté des administrés, on distinguera les citoyens mais

[1] http://www.help.gv.at/ (dernier accès : 25 sept. 08)
[2] http://www.bhoomi.kar.nic.in/ (dernier accès : 25 sept. 08)
[3] http://www.floodforum.org.uk/ (dernier accès : 25 sept. 08)
[4] https://www.amendes.gouv.fr/ (dernier accès : 25 sept. 08)

également les représentants d'entreprises et d'associations. Globalement, il s'agit d'un public très varié, en particulier pour ce qui concerne les citoyens à titre privé, de tous âges, toutes compétences en informatique, et toutes conditions physiques. L'utilisateur est souvent tenu de réaliser la procédure dématérialisée sur l'application, sous peine d'amende, de poursuite, de cessation de ses droits ou de ceux d'un tiers. L'administré utilisateur d'une application d'e-Procuration doit être en situation de confiance envers l'application pour accepter de fournir des informations privées voire confidentielles, et croire en la validité de la réalisation de sa procédure administrative de façon dématérialisée. Les agents administratifs quant à eux utilisent les applications d'e-Procuration pour leur activité de travail quotidienne : les applications d'e-Procuration doivent fournir un support adapté à leurs habitudes, leur jargon et leur environnement.

L'acceptabilité d'une application d'e-Procuration est cruciale pour ses utilisateurs mais aussi pour les institutions qui y sont engagées : en cas d'incident, l'image de l'institution en pâtit, des poursuites peuvent être engagées et la confiance des utilisateurs est altérée. Pour assurer tous les aspects liés au développement de ces applications, différents corps de métier interviennent lors de la conception de l'interface sur le plan technique (analyse fonctionnelle, gestion éditoriale, analyse de l'activité, développement, ergonomie, bases de données, graphisme, sécurité, réseaux) mais aussi sur le plan administratif (conduite de projet, consultation des institutions concernées, aspects légaux). Parmi les défis que doivent relever les équipes multidisciplinaires de concepteurs d'applications d'e-Procuration, on remarquera les besoins suivants : fournir des services fonctionnels et procéduraux complexes susceptibles de subir des modifications régulières en cas de nouvelle législation, stocker et manipuler de grands volumes de données personnelles de façon sécurisée, et plus particulièrement produire des interfaces utilisables sur lesquelles repose, généralement, un grande partie de la crédibilité accordée par les utilisateurs aux applications. Une plateforme de communication pour ces intervenants dans le processus de conception est donc essentielle, pour la cohérence et la satisfaction de ces besoins de différentes natures. Pour un projet de conception en e-Gouvernement incluant plusieurs partenaires, la communication est reconnue par les praticiens eux-mêmes comme le facteur le plus important de la réussite : établissement d'objectifs clairs, constitution de communautés de conception actives et partageant leurs expériences (Halstead-Nussloch et coll. 2003).

L'objectif principal de ce travail de thèse est de fournir un cadre méthodologique pour la conception d'interfaces d'applications d'e-Procuration. Pour cela, nous nous intéressons aux processus de conception d'interface. Notre objectif est de fournir les supports les plus adaptés pour que l'utilisabilité soit prise en compte par tous les intervenants dans le processus de développement. Une étude approfondie du contexte a permis de dégager les besoins suivants, pour une méthode de conception d'applications d'e-Procuration :

- Développement d'applications utilisables, en accord avec les besoins des utilisateurs et leurs compétences, si inégales soient-elles ;
- Capitalisation des bonnes conduites de conception d'interface, avec une visée d'harmonisation des interfaces du domaine à plus long terme ;
- Clarification de la communication au sein de l'équipe de conception, pour une spécification de l'interface non ambiguë et lisible par chacun des corps de métier impliqués dans la conception en e-Procuration ;

La réflexion est menée, d'une part, sur la base d'une étude théorique recherchant parmi les méthodes de conception existantes celles qui seraient applicables à notre domaine d'étude. D'autre part, nous nous sommes appuyés sur des observations exploratoires menées sur le terrain, au sein d'une entreprise de technologies de l'information chargée de la conception d'applications d'e-Gouvernement. Au vu des activités réelles et des besoins des concepteurs, la diffusion de recommandations ergonomiques lisibles et concrètement applicables s'avère opportune pour que les interfaces conçues soient utilisables.

Dans cette thèse, nous proposons une méthode de conception pour l'e-Gouvernement basée sur des patrons de conception dédiés aux interfaces. La définition de patron employée est « une solution à un problème de conception récurrent, qui a été prouvée scientifiquement ou empiriquement : sa validité a quoiqu'il en soit été éprouvée par la pratique, et la communauté reconnaît le bien-fondé de son raisonnement ». Cette solution se veut concrète et directement applicable à un problème rencontré, moyennant une adaptation au contexte concerné. Cette méthode de support à la conception a été initiée dans le domaine de l'architecture, au travers de patrons de conception de bâtiments, de rues ou encore de jardins publics (Alexander et coll. 1977). En informatique, le problème traité par un patron de conception peut porter sur l'architecture logicielle (Gamma et coll. 1994), l'organisation de l'équipe de conception, ou encore l'interface utilisateur. C'est à ce dernier type de patrons de conception que nous nous intéressons ici : un patron d'interface répond à un problème de conception d'interface sur la base de la connaissance de la communauté en Interaction Homme-Machine (IHM), et, en fournissant une solution directement applicable, guide le concepteur vers de bonnes pratiques de conception. Dans ce mémoire, les patrons de conception d'interfaces pour les applications d'e-Procuration sont décrits de façon systématique à partir de fragments d'interfaces récurrents listés à partir d'une analyse de l'existant.

La méthode de conception que nous proposons est basée sur la pratique de prototypage. Les patrons servent de support à l'analyse et au développement d'une application d'e-Procuration en guidant les concepteurs vers les solutions les plus appropriées. L'organisation de patrons est étudiée en détails, pour favoriser la navigation en leur sein et la recherche clairvoyante de solutions tout au long du processus de conception. Un outil soutient l'utilisation de cette méthode, et fournit un guidage pour la définition de la structure hypertexte de l'application à concevoir. Cette méthode de conception basée sur patrons a été employée dans l'industrie sur des projets réels et son applicabilité à été évaluée sur le terrain, auprès d'utilisateurs réels, quelques mois après son déploiement. Cette évaluation nous a permis d'évaluer des hypothèses de recherche concernant le support à l'analyse fonctionnelle, le guidage dans la conception d'interfaces ergonomiques, la capitalisation et la mise à disposition de la connaissance en conception d'interfaces. Cette évaluation a également ouvert des pistes d'amélioration pour une meilleure adaptation aux conditions de travail dans une entreprise et une meilleure acceptabilité par ses utilisateurs. Ce mémoire de thèse présente ces réflexions et contributions, et s'articule de la manière suivante :

- Le chapitre 2 fait un état de l'art des cycles de vie et des méthodes de conception d'applications interactives. La place accordée à l'utilisateur dans le processus de conception, les moyens offerts pour spécifier l'application, puis pour évaluer son utilisabilité seront étudiés plus particulièrement. Les critères évoqués précédemment seront précisés et leur couverture par chaque méthode étudiée sera évaluée.
- Le chapitre 3 présente l'observation exploratoire que nous avons menée sur le terrain d'une entreprise de technologies de l'information dédiée à l'e-Gouvernement en octobre 2006. L'étude porte sur les activités de conception, et sur une analyse des interfaces des applications développées par l'entreprise.
- Le chapitre 4 propose une méthode de conception d'interfaces pour l'e-Gouvernement basée sur des patrons. Forte des observations réalisées dans le chapitre 3, notre méthode se fonde sur les principes de standardisation et de capitalisation de la connaissance de conception, particulièrement adaptés à l'e-Gouvernement.
- Le chapitre 5 présente l'évaluation d'une méthode de conception d'interfaces basée sur des patrons, développée et déployée en contexte industriel par l'entreprise qui avait accueilli l'observation exploratoire du chapitre 3. Une observation de faits provoqués et des entretiens avec des analystes utilisateurs de cette méthode nous a permis de valider des hypothèses de recherche.

- Le chapitre 6 présente l'outil eGovIPM que nous avons développé en guise de support à la méthode présentée dans le chapitre 4 et qui garantit sa mise en œuvre concrète. Il guide le concepteur dans l'édition de la structure hypertexte de l'application, sur la base de patrons d'enchaînement d'écrans.
- Le chapitre 7 discute des apports et des limites de ce travail, et ouvre des perspectives pour des travaux futurs.

2. État de l'art des méthodes de conception IHM

Résumé

De nombreux cycles et méthodes de conception guidant les étapes nécessaires à la réalisation au déploiement d'une application informatique interactive ont été développés. Cette section étudie les cycles de développement existants puis les méthodes de support à la conception susceptibles de couvrir les besoins du domaine de l'e-Gouvernement : 1) prise en compte de l'utilisateur, de ses besoins et de ses contraintes, 2) spécification de l'application sous plusieurs angles de vue pour une couverture complète et cohérente des besoins, 3) évaluation de l'utilisabilité en cours de conception et après le déploiement de l'application, 4) gestion et diffusion de la connaissance ergonomique pour un guidage de la conception vers une application utilisable. Cette étude révèle notamment la pertinence d'une organisation de la connaissance ergonomique sous forme de patrons d'interfaces pour la conception en e-Gouvernement.

Une application d'e-Procuration est avant tout un système interactif et sa conception requiert un cadre méthodologique tel que ceux que proposent des cycles de développement déjà éprouvés par de nombreux et divers domaines d'application. L'e-Gouvernement a ses propres contraintes légales, techniques et éthiques : nécessité de fournir un guidage à l'utilisateur ; public très large d'utilisateurs, y compris des personnes dont la vue ou la gestuelle sont altérées (p.ex. personnes âgées, mal voyants) ; cohérence de la définition de la procédure administrative avec sa dématérialisation sur le Web ; synchronisation des activités et gestion des rôles pour la réalisation de la procédure ; garantie de la sécurité du transfert, du stockage et du traitement des données. De plus, plusieurs types d'intervenants participent à la conception d'une application d'e-Procuration : une équipe technique, mais aussi des responsables d'institution et parfois des utilisateurs finaux, citoyens ou agents administratifs. Dans ce chapitre, une étude des cycles et des méthodes de conception existants évalue leur adéquation aux exigences de la conception en e-Gouvernement. Assurent-ils que l'application en cours de conception sera conforme aux besoins et aux capacités de l'utilisateur ? Favorisent-ils et clarifient-ils la communication au sein d'une équipe de conception pluridisciplinaire ? Offrent-ils des moyens de capitaliser la connaissance acquise au cours des expériences de conception ?

Les cycles de développement sont étudiés d'abord (§2.1) : ils définissent un enchaînement synchronisé d'activités pour la conception. C'est un cadre méthodologique générique qui doit être ajusté au domaine d'application et aux besoins inhérents à ce domaine. La réalisation concrète d'une phase nécessite ainsi un outillage adapté, en termes de formalisme de description ou encore de méthodes de communication et de réflexion. Lorsqu'un intérêt tout particulier est porté sur la conception de l'interface utilisateur, des méthodes sont disponibles, issues de la recherche en IHM et en génie logiciel, pour orienter la conception vers l'utilisateur final, dès les premières phases de développement (§2.2).

La spécification du système interactif est aussi une activité de conception fondamentale. L'utilisation de méthodes de support à la spécification doit être souple et rapide afin d'encourager le développement de multiples alternatives. Elle doit également être en équilibre entre le formel et l'intuitif, pour que la spécification soit à la fois non ambiguë et lisible. Plusieurs points de vue doivent figurer dans la spécification d'une application d'e-Procuration, de par sa nature d'application Web fonctionnelle déployée dans un contexte soumis à de fortes contraintes procédurales et organisationnelles (§2.3).

En dépit de l'intérêt porté à l'utilisateur et de l'étude attentive de ses besoins, des problèmes d'utilisabilité peuvent survenir lors de la confrontation du système à ses utilisateurs. Des

méthodes d'évaluation permettent de valider certains aspects de l'utilisabilité du système dès sa conception ou après son déploiement, en présence ou non d'utilisateurs finaux (§2.4). La connaissance ergonomique pour la conception de systèmes interactifs utilisables fait l'objet d'une capitalisation formelle et systématique de sorte qu'elle soit exploitable par les concepteurs, et ce sous plusieurs formes : bonnes pratiques, recommandations, ou encore points décisifs sur lesquels les concepteurs doivent porter leur vigilance (§2.5).

2.1. CYCLES DE DÉVELOPPEMENT DE SYSTÈMES INTERACTIFS

Depuis les années 1970, et le développement alors exponentiel d'applications informatiques interactives tant dans le domaine privé que professionnel, des travaux de recherche ont été menés activement pour offrir aux concepteurs de ces applications des cadres méthodologiques de soutien à leur activité. Ces cycles de développement sont des enchaînements recommandés de phases de conception, parfois couplés à des langages de modélisation pour la documentation du projet. Ils ont fourni des moyens de limiter les échecs de projets de conception informatiques, en optimisant la coordination des ressources des équipes de conception, en se préoccupant de l'acceptabilité de l'application ou encore en réduisant la portée des conséquences d'une erreur de jugement ou d'une faille technique.

2.1.1. Cycle en cascade (Royce 1970)

Le processus de développement d'applications interactives proposé par Winston Royce est une séquence d'activités dont l'enchaînement ordonné permet à chaque activité de profiter des réflexions et des artefacts produits par la précédente. Ce processus est traditionnellement schématisé par des activités enchaînées en cascade (cf. Figure 1).

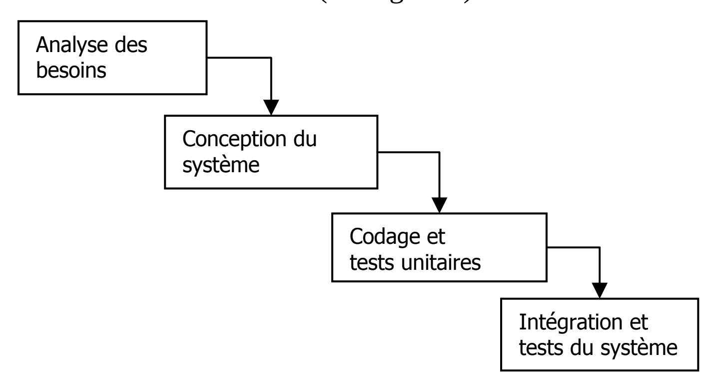


Figure 1. Cycle de développement en cascade initial

Des problèmes d'utilisabilité, des problèmes techniques ou des réajustements nécessaires dans la définition des besoins peuvent émerger relativement tard dans le cadre de ce processus. Cela peut se produire au moment de la confrontation des utilisateurs avec l'application, au cours des tests finaux, voire après le déploiement de l'application. Des retours sont alors proposés vers des phases antérieures dans le processus (p.ex. retour vers le codage en cas de bug informatique sur une fonctionnalité qui serait détecté après déploiement). Si un retour en arrière est nécessaire vers une phase de début de cycle, le coût peut être élevé pour la conception. Les versions récentes du cycle de vie en cascade font apparaître à chaque étape une activité de vérification (McConnell 1996) : la gestion de projet doit garantir la validité du passage d'une étape à l'autre. Le cycle de vie en cascade est aujourd'hui utilisé dans des projets de très grande envergure (p.ex. *Department of Defense*[1] aux États-Unis ou la NASA *National Aeronautics and Space Administration*[2]) pour lesquels une spécification des besoins solide et immuable est nécessaire, rendant peu probable la redéfinition de ceux-ci et plus généralement les retours vers des étapes antérieures.

[1] Département de la Défense, chargé de la coordination de la sécurité nationale et militaire
[2] Administration nationale de l'aéronautique et de l'espace

L'intégration des problématiques de conception centrée utilisateur est envisageable dans un tel contexte (Curtis et Hefley 1994). Chaque étape du cycle de développement en cascades est considérée sous deux points de vue : l'interface et l'application comme le montre la Figure 2. Les études et développements inhérents à ces deux facettes sont à explorer, ce qui pousse les concepteurs à prendre l'utilisabilité de l'application en considération. La dichotomie entre ces deux points de vue s'avère toutefois très tranchée, de l'aveu même des auteurs de la méthode, ce qui impose une rigidité fâcheuse pour les évolutions de l'application. En effet, les modifications impliquent généralement à la fois des aspects interface et fonctionnel.

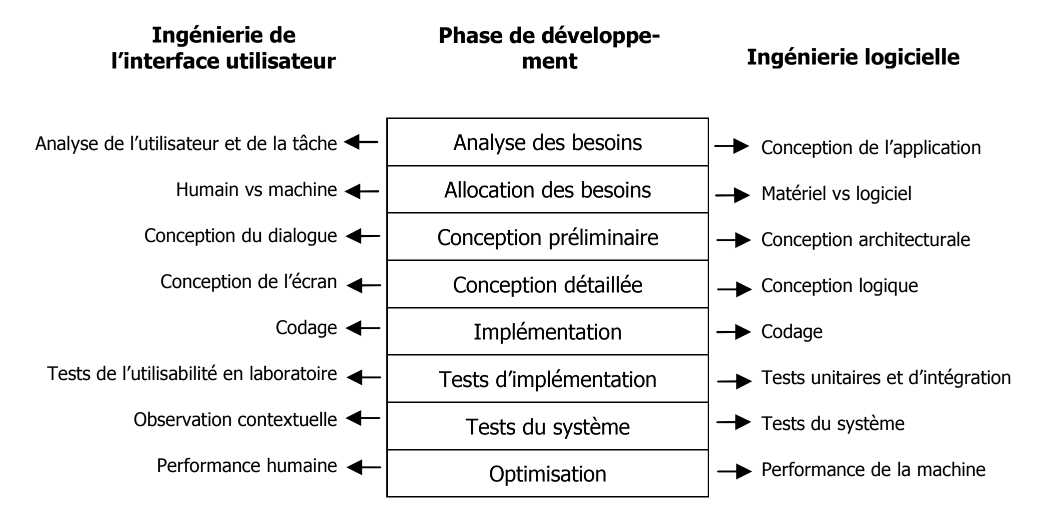


Figure 2. Intégration de l'interface dans un processus de conception en cascade (issu de (Curtis et Hefley 1994))

2.1.2. Cycle en V (McDermid et Ripken 1984)

Le cycle de développement en V (cf. Figure 3) détaille le processus de développement d'une application interactive de telle sorte qu'à chaque étape productrice d'artefact (document, module d'application ou combinaison de modules vers l'application finale) corresponde une étape qui évalue cet artefact. Trois catégories d'étapes se dégagent:

- Étapes dans la phase descendante (le bras gauche du V) qui couvre toutes les étapes de raffinement des besoins jusqu'à l'implémentation de l'application. Les étapes de conception sont détaillées en fonction de la granularité des problèmes considérés, des plus abstraits aux plus concrets ;
- Codage de l'application (la base du V) qui vient suite à cette phase de conception ;
- Étapes dans la phase ascendante, ensuite (le bras droit du V), chacune correspondant à un contrôle d'une étape de la phase descendante.

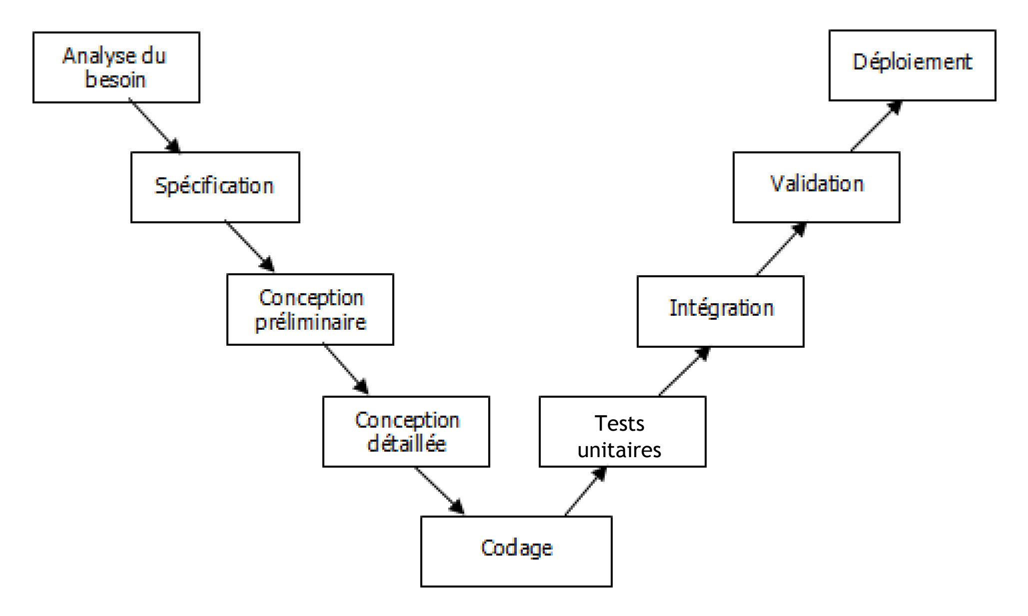


Figure 3. Cycle de développement en V

A chaque étape de conception correspond ainsi une étape de vérification. Certains tests de vérification sont réalisables directement lors de l'étape de conception qui leur correspond, mais la plupart des tests a lieu après la phase de codage, conformément au sens de lecture du cycle en V. Il est à noter que les tests à effectuer dans la phase ascendante sont à spécifier dès l'étape de phase descendante associée. Par exemple, en phase de *Conception préliminaire*, l'équipe de conception effectue les opérations inhérentes à cette étape et, en parallèle, elle définit les moyens de vérifier que l'application finale sera conforme aux besoins ayant émergé de cette phase de conception. Cette vérification aura lieu en phase de test correspondant, c'est-à-dire lors de l'*Intégration*. Pour les premières phases, une longue période de temps s'écoule donc entre la phase de conception et la phase de test correspondante, si elle a lieu seulement lors de la phase ascendante. En cas de correction à effectuer, il y a un retour vers le début du cycle, et un réajustement nécessaire des phases suivantes, ainsi que des batteries de tests déjà spécifiées pour la phase de validation.

La flexibilité et la large renommée du cycle en V font qu'il est utilisé dans de nombreuses entreprises de développement informatique. La documentation nécessaire pour la mise en correspondance d'une phase de production (ligne descendante) et de la phase de vérification qui lui est associée (ligne ascendante) sert à la gestion de projet et à la cohérence de la conception, y compris dans les projets de grande envergure.

2.1.3. Cycle en spirale (Boehm 1986)

Le cycle de développement d'applications interactives en spirale (cf. Figure 4) est aussi appelé cycle de développement itératif. Il correspond en effet à plusieurs itérations sur quatre types d'étapes dont le contenu évoluera au fil des itérations, mais dont la nature reste la même :

- Définition des objectifs, des alternatives possibles et des contraintes du projet ;
- Évaluation des alternatives en réponse aux besoins et en tenant compte des contraintes : cette phase comporte un travail de prototypage, et le prototype sera de plus en plus opérationnel au fur et à mesure des itérations ;
- Conception d'une solution apparaissant comme la meilleure des alternatives étudiées en phase précédente : cette phase englobe les étapes de conception et de vérification déjà observées dans les cycles de développement en cascade et en V (spécification, conception préliminaire puis conception détaillée, tests unitaires puis tests d'intégration, déploiement) ;

- Planification de la phase suivante, et notamment affectation des tâches à réaliser au sein de l'équipe de conception.

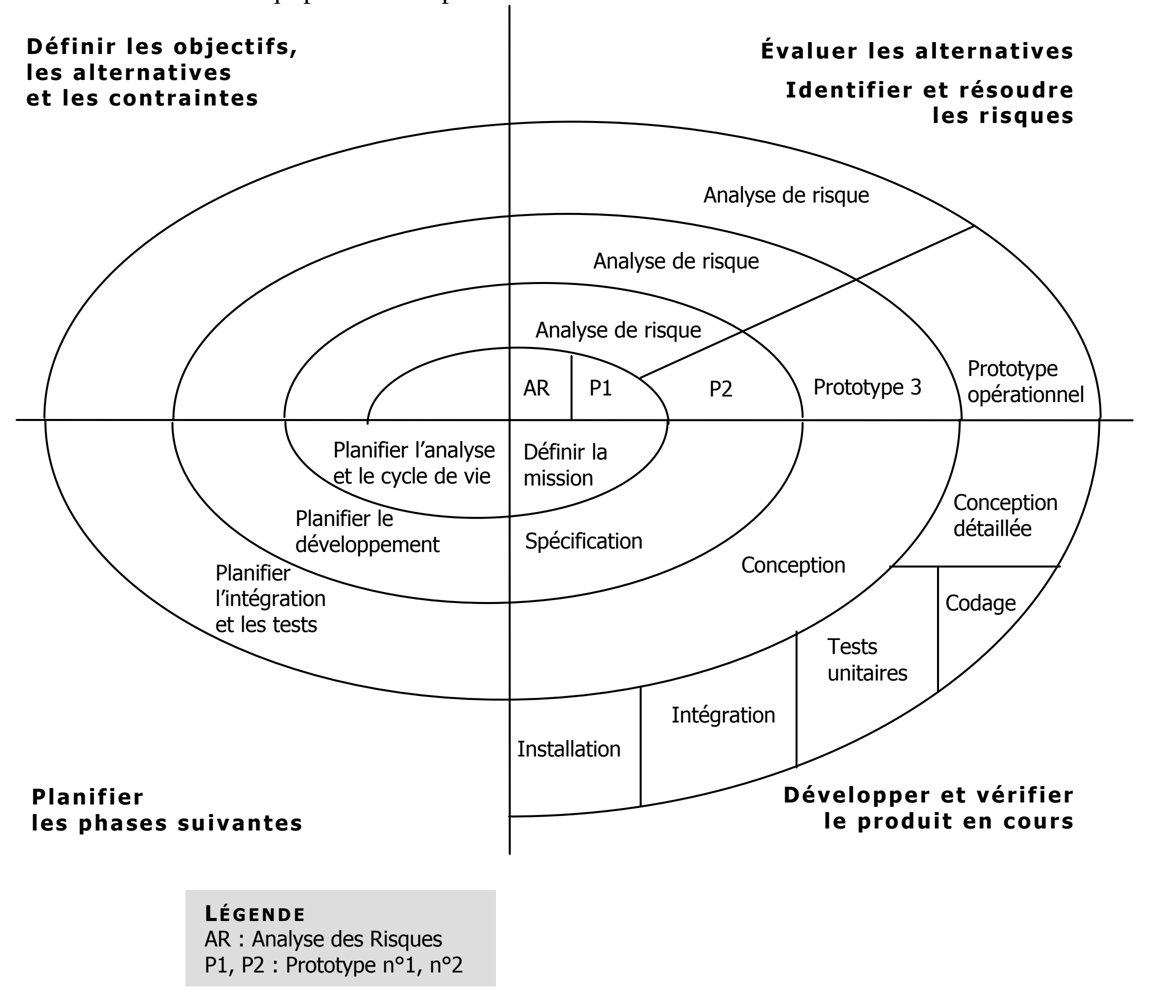


Figure 4. Cycle de développement en spirale

Ce cycle de développement permet ainsi d'explorer plusieurs alternatives, en intégrant éventuellement une activité de maquettage. Il s'agit en réalité de travailler itérativement à la définition des besoins et à la spécification de l'application. Une fois que la réflexion a atteint un degré de raffinement permettant la prise de décision, l'équipe de conception enchaîne avec un cycle de développement en cascade ou en V pour mener à bien les phases concernant la conception détaillée puis l'implémentation proprement dite. Ce cycle de développement est donc long et coûteux à mettre en place, et requiert en particulier que les personnes chargées de la gestion du projet soient formées à un tel processus et à une gestion des ressources qui s'adapte à la morphologie de ce genre de projet.

2.1.4. Cycle *Star*[1] (Hix et Rex Hartson 1993)

Ce cycle de développement a été établi pour le développement d'interfaces utilisateurs, en réponse aux cycles de développement d'applications préexistants. Deborah Hix considère en effet que la conception et le développement d'interface ne suivent pas des parcours classiques de réflexion, parcours analytiques ou synthétiques également appelés *top-down* ou *bottom-up*[2]. Il est constaté dans ces travaux que la conception d'interfaces n'est pas une activité séquentielle et que les tâches de conception peuvent être réalisées dans n'importe quel ordre, chacune d'elles étant évaluée au fur et à mesure. Le prototypage est un moyen d'évaluer la validité de

[1] Étoile
[2] De haut en bas ; de bas en haut

l'interface, par le biais d'expertises ou bien de tests utilisateurs, même si l'application est encore en cours de développement. Comme il est indiqué sur la Figure 5, l'évaluation est ainsi au centre de ce cycle de développement.

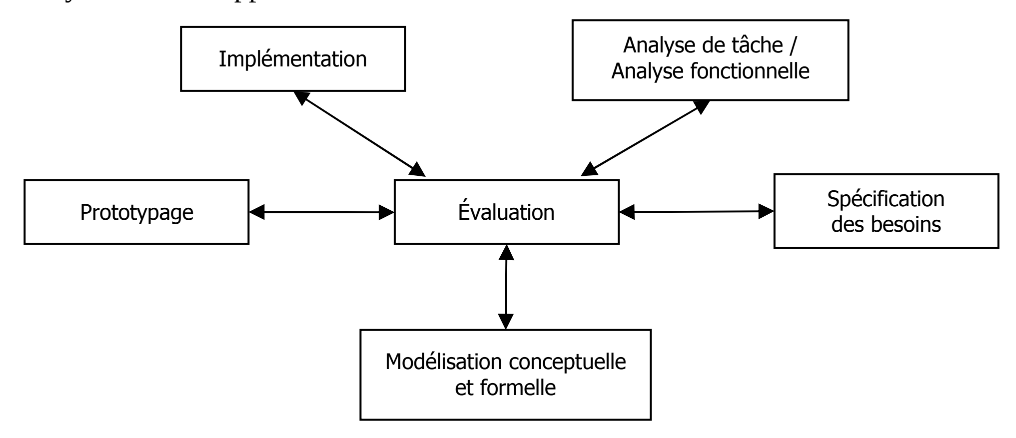


Figure 5. Cycle de conception d'interfaces Star

Toutes les tâches de conception sont très interconnectées, et reliées entre elles par la tâche d'évaluation : chaque tâche de conception est ainsi validée dès la fin de sa réalisation, avant de passer à une autre tâche. Théoriquement, la conception d'interfaces peut commencer à n'importe quelle phase du cycle Star, chaque phase alimentant, via les résultats de son évaluation, les réflexions menées dans les autres phases. Il s'agit là d'une très grande liberté laissée à l'équipe de conception et les premières itérations d'un tel cycle seront difficiles à dépouiller et à évaluer. Si une partie de l'équipe commence par une phase, une autre partie par une autre phase, aucun guidage n'est fourni pour trancher entre les alternatives trouvées par chaque partie dans le cas où elles ne seraient pas modulables, pour des raisons techniques ou fonctionnelles. Le cycle de conception Star convient donc aux équipes de conception de petite taille, où chaque composante a une excellente conscience de l'avancement des activités des autres.

2.1.5. Cycle *Object-Oriented User Interface*[1] (Collins 1995)

Dave Collins propose OOUI, une approche de développement des interfaces utilisateurs basée sur les principes des méthodes orientées objet. L'approche guide le développement des interfaces présentées à l'utilisateur mais aussi de l'épine dorsale programmatique qui leur est associée. L'objectif est de fournir une vision de l'application qui soit compréhensible et discutable par tous les intervenants au processus de conception. Comme le montre la Figure 6, les activités de conception de l'interface sont réalisées en parallèle d'activités de conception de contrôle de l'application, en réaction aux aspects interactifs. Cette méthodologie suit de plus les préceptes MVC (Modèle Vue Contrôleur) (Reenskaug 1979) pour les liens entre modèles de l'interface et de l'application.

Les utilisateurs sont considérés de façon centrale et complexe dans la méthodologie, selon leurs activités ou encore leurs modèles cognitifs et culturels. Il est préconisé par OOUI d'intégrer les utilisateurs dès les premières phases de conception, et c'est pour recueillir leurs opinions que le prototypage est employé. L'emploi de modèles de tâches et l'étude de l'activité sont des phases intéressantes du point de vue de la conception centrée utilisateur ; toutefois, l'implication de leur interprétation sur les autres modèles de conception est peu guidée.

[1] Interface utilisateur orientée objet

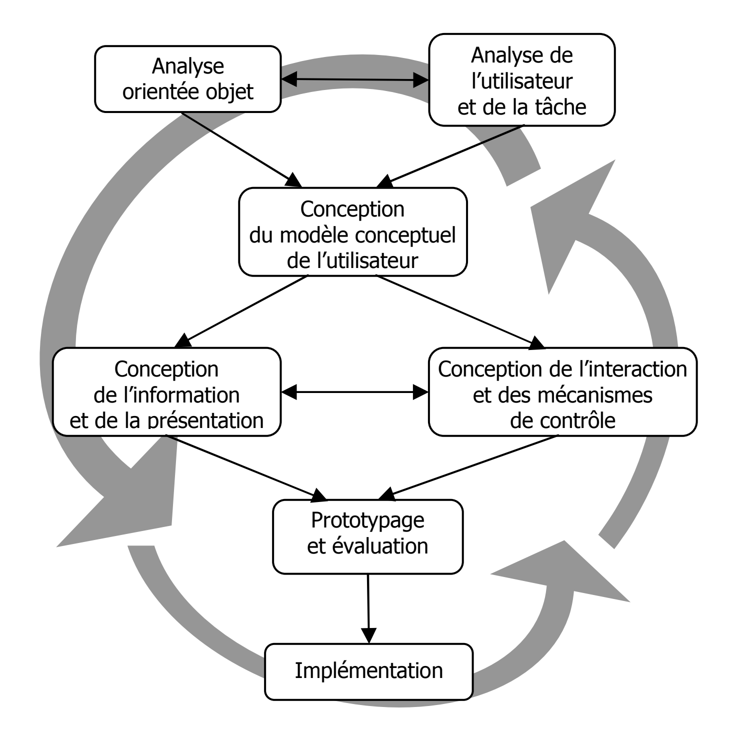


Figure 6. Cycle de développement OOUI (Collins 1995)

2.1.6. Cycle ∇ (Nabla) (Kolski 1998)

Ce cycle de développement vise à intégrer les phases de prise en compte de l'utilisateur aux cycles de développement traditionnels du domaine du génie logiciel. Nabla possède une représentation graphique inspirée du cycle de vie en V (cf. §2.1.2).

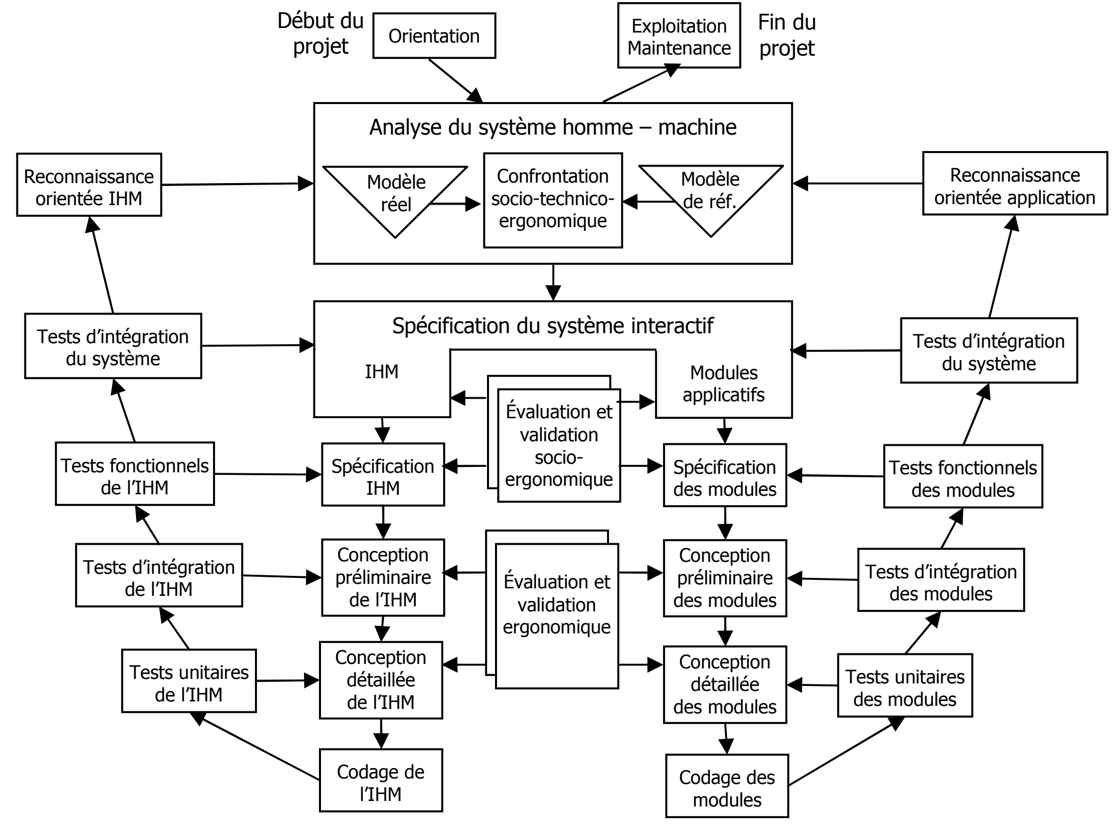


Figure 7. Cycle de développement Nabla (issu de (Kolski 1998))

La conception commence à la verticale des deux branches du V, par l'étude approfondie des interactions de l'application. S'ensuivent les spécifications de l'interface et de l'application, ponctuées de vérifications de leur validité, et de leur cohérence entre elles. Puis, les étapes de conception d'interface sont détaillées dans le bras gauche du V ; celles de la conception de l'application elle-même sont détaillées dans le bras droit du V. Un retour sur les besoins en interaction advient à l'issue de ces étapes. Si les besoins évoqués en début de conception aux niveaux sociaux, techniques et ergonomiques sont satisfaits, l'application peut être déployée.

Comme le montre la Figure 7, les propositions en matière de spécification sont évaluées régulièrement, de part et d'autre du V, ce qui permet de garantir une cohérence entre les documents de conception de l'interface et ceux de l'application. Il est toutefois regrettable que la modélisation de la tâche de l'utilisateur ne soit pas plus explicitement intégrée à la conception de l'interface (Lepreux, Abed, et Kolski 2003) dans un cycle de développement attentif à intégrer les aspects utilisateurs.

2.1.7. *Rational Unified Process*[1] (Kruchten 2000)

Le processus de développement RUP est un cadre méthodologique pour le développement d'applications interactives, s'appuyant notamment sur la méthodologie UML (Rumbaugh, Booch, et Ivar Jacobson 1999). UML fournit en effet un ensemble de modèles permettant la représentation de processus, tant au niveau de l'application (p.ex. représentation des besoins des différentes catégories d'utilisateurs vis-à-vis de l'application) qu'au niveau du travail de conception lui-même (p.ex. représentation des compétences de chaque intervenant). Comme le montre la Figure 8, un cycle de développement RUP consiste en une séquence incrémentale d'itérations pour aboutir à l'application interactive finale. Chaque itération consiste en un ensemble de tâches appartenant à des disciplines diverses, tâches d'ingénierie et tâches de gestion de projet. Une itération entre dans le déroulement d'une phase, et parfois plusieurs itérations sont nécessaires pour que la phase soit validée. Quatre phases se succèdent dans la vie d'un projet qui suit la méthodologie RUP :

- Lancement : compréhension de la portée du projet, de ses coûts et de sa rentabilité. En parallèle, une analyse des besoins utilisateurs est menée. A l'issue de cette phase, la décision est prise de continuer le projet ou de l'abandonner.
- Élaboration : un cas d'utilisation complet à 80% est développé pour que l'architecture de l'application soit cernée dans sa globalité et qu'elle soit validée. Un prototypage rapide est souvent réalisé pour mieux appréhender chacun des problèmes techniques ayant déjà émergé des réflexions. Il arrive qu'à l'issue de cette phase, un projet soit abandonné sans que les conséquences pour les partenaires soient désastreuses.
- Construction : une première version opérationnelle de l'application est construite à partir des résultats de la phase précédente. La gestion des ressources entre alors en ligne de compte, et l'implémentation est réalisée.
- Transition : l'équipe de conception vérifie que l'application développée correspond bien aux exigences de ses utilisateurs finaux et des partenaires impliqués dans la conception. L'application, une fois validée, est déployée dans l'environnement de ses utilisateurs finaux.

[1] Processus rationnel unifié

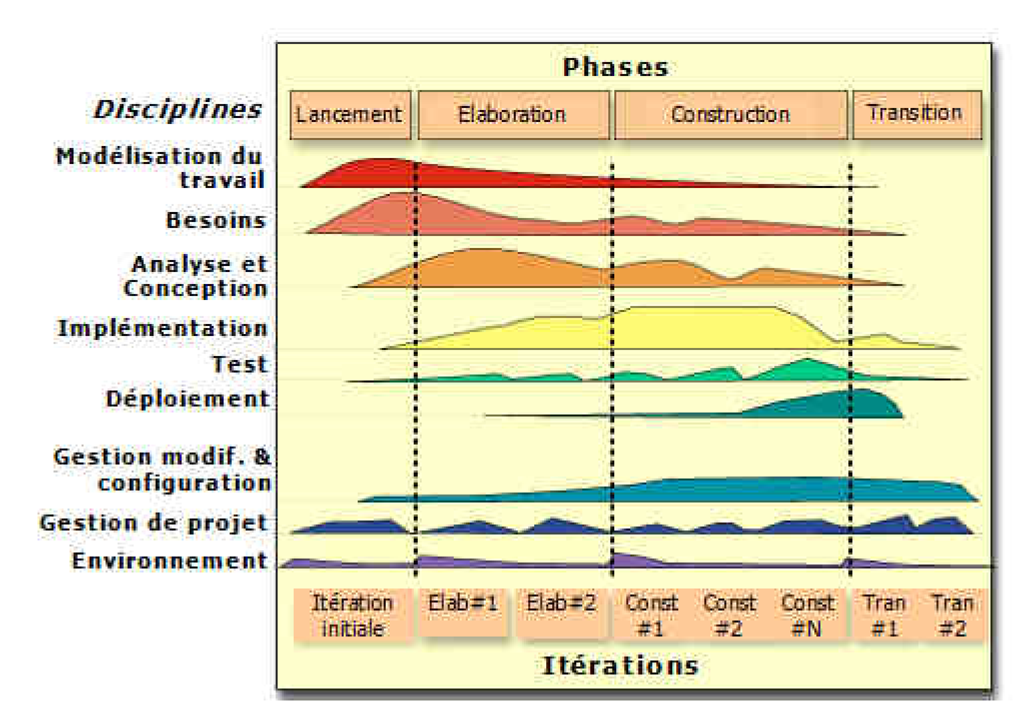


Figure 8. Cycle de développement RUP

Sur la base des travaux réalisés par Barry Boehm pour le cycle de développement itératif (cf. §2.1.3), RUP propose un cadre pour la gestion de projet et le développement d'une application interactive. RUP considère ainsi avec soin l'adaptation aux contraintes matérielles du projet, la direction du travail collaboratif par une gestion pertinente des compétences ou encore le découpage de problèmes de conception en problèmes de plus petite taille qui puissent être traités en parallèle pour optimiser le temps alloué à la conception. D'autre part, la vision itérative et incrémentale autorise une prise de décisions de conception pas à pas, avec éventuellement des retours en arrière. Toutefois, la lisibilité globale du processus de conception risque de se perdre dans les itérations successives, d'autant que sur un projet de grande ampleur, les équipes travaillent en parallèle. Pour gérer une cohésion globale, et organiser les ressources du projet, une solide formation à la méthodologie RUP est requise pour l'équipe de gestion de projet, en plus de la formation nécessaire, pour chaque intervenant, à la lecture et à l'édition de modèles UML pour la documentation du projet.

2.1.8. Cycle de vie en O des applications Web (Scapin et coll. 2000)

Les applications Web sont des applications interactives se distinguant fortement les unes des autres par leur nature (p.ex. blog personnel, site de vente en ligne ou vitrine d'une association), le nombre de pages qu'elles contiennent, la complexité des fonctionnalités qu'elles proposent (certaines requérant une communication avec un système tiers), le nombre de visites qu'elles reçoivent, le rythme des mises à jour (p.ex. quotidien pour des résultats sportifs, annuel pour le site Web d'un hôtel de petite taille) ou encore leur durée de vie. La diversité des applications Web fait que de nombreux cycles de vie peuvent répondre aux contraintes de ces projets. Pour le moment, il n'existe pas de consensus, ni sur les phases de développement nécessaires, ni sur le cycle de vie qui décrirait au mieux le processus de développement des applications Web. Toutefois le cycle de vie du développement d'un site Web peut généralement être vu comme un processus itératif. Le cycle de vie en O est ainsi un processus de conception itératif pour le Web, composé des six phases suivantes (voir Figure 9) :

- Expression des besoins (c'est-à-dire Analyse des besoins). Cette phase identifie les principaux buts des commanditaires, le contexte d'utilisation et les besoins. Elle comprend la collecte du contenu de l'application, qui sera utilisé plus tard.

- Spécification du site (c'est-à-dire Modélisation conceptuelle). Cette phase produit des spécifications à partir du contexte d'utilisation et des besoins recueillis dans la phase précédente. Des modèles détaillés sont construits pour formaliser les besoins, comme, par exemple, les tâches utilisateur avec l'architecture de l'application et du site ;
- Conception du site. Cette phase consiste à affiner les spécifications d'après leur contenu. A la fin, une carte de navigation et un modèle de page ont été définis, ainsi qu'une spécification détaillée qui guidera l'implémentation de l'application Web ;
- Développement du site (c'est-à-dire Implémentation). Cette phase correspond à la construction physique de l'application Web, à la production des pages HTML et à l'éventuelle intégration de lecteurs de média (son ou vidéo). A la fin, toutes les pages ont un contenu, des liens et des éléments graphiques incorporés, et l'application est livrée ;
- Utilisation et évaluation du site. Cette phase est destinée à évaluer des prototypes avancés avec des utilisateurs finaux. Elle a aussi pour but de vérifier que le produit des phases précédentes respecte les besoins et le contexte d'utilisation identifiés dans la première phase. L'évaluation de l'utilisabilité, par des tests utilisateurs ou par l'inspection de règles ergonomiques, est l'activité principale de cette phase.
- Maintenance du site. Cette phase peut durer longtemps puisque c'est ici que le concepteur est chargé de recueillir de nouvelles informations et les modifications de planning qui ont été demandées au cours de la phase d'utilisation et d'évaluation.

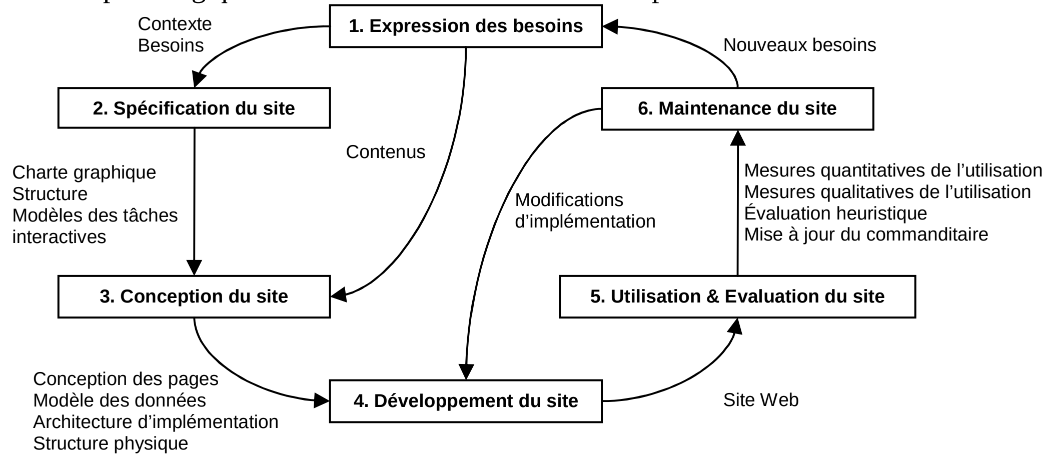


Figure 9. Cycle de vie en O d'un site Web (Scapin et coll. 2000)

Ce processus cyclique ne prend pas en compte les activités de prototypage qui sont habituellement le coeur du processus de conception des applications Web. C'est pourquoi deux flèches ont été ajoutées au milieu de la boucle. Dans la Figure 9, la flèche de gauche (intitulée « Contenu ») représente le possible raccourci dans la phase de spécification. En effet, au début du projet, le concepteur peut commencer immédiatement la conception du site pour pouvoir discuter avec les commanditaires à partir d'informations précises. La flèche de droite (intitulée « Modifications d'implémentation ») représente un possible raccourci pour que le développement soit plus rapide et pour tenir compte de l'utilisation et des évaluations de manière plus centrale. En effet, le développeur de l'application Web peut modifier directement l'application après une évaluation, sans repasser par les trois premières phases du processus de conception.

2.1.9. Cycles de développement pour l'e-Gouvernement

Le cycle de développement d'une application d'e-Procuration doit fournir un cadre qui permette de s'assurer de la cohérence de l'application avec la procédure qu'elle dématérialise. Cette cohérence doit être maintenue tout au long de la vie de l'application. Les cycles itératifs,

tels que le **cycle en spirale** (cf. 2.1.3), permettent de considérer tous les aspects de l'application en parallèle, partant d'une description sommaire qui s'enrichit au fil des itérations. Cette articulation des activités de conception permet de gérer les contraintes complexes de l'e-Procuration, qui portent à la fois sur l'interface, la procédure, la gestion des données, la sécurité, et aussi les enjeux politiques sous-jacents. Un séjour dans l'entreprise SmalS (cf. chapitre 3) chargée de la conception d'applications d'e-Gouvernement en Belgique nous a permis d'observer le cycle de développement suivi : il s'agissait d'un cycle en spirale. Une attention particulière y est portée à l'interface, le prototypage est au centre du développement, et le prototype est enrichi jusqu'à devenir l'interface finale. C'est alors que l'application *back-end* est développée.

Des modifications sont à prévoir sur l'application après son déploiement, dès qu'une nouvelle législation modifie la définition de la procédure. Le **cycle de vie en O** (cf. §2.1.8) considère ces aspects évolution post déploiement avec finesse : les évolutions majeures, nécessitant une correction des documents de conception, sont traités différemment des plus mineures, concernant simplement l'implémentation.

Plusieurs expériences d'utilisation du cycle de développement **RUP** (cf. §2.1.7) pour des applications d'e-Gouvernement figurent dans la littérature. Pour les applications de son portail gouvernemental inforME[1], l'état du Maine (États-unis d'Amérique) préconise un processus de conception dont la séquence d'activités est adaptée d'un plan RUP[2]. L'entreprise EzGov[3] suivait également une méthodologie RUP pour la conception d'applications d'e-Gouvernement mais cette méthode s'est avérée insuffisante pour gérer la lisibilité des compétences en conception d'interface et a rendu complexe la communication au sein de l'équipe de développement (Boersma 2005). Les artefacts manipulés, issus des recommandations RUP, étaient interprétés différemment par les diverses catégories de concepteurs. Cette ambiguïté posait des problèmes de communication : une redéfinition formelle des artefacts manipulés a permis de clarifier le processus de développement et d'établir un vocabulaire commun.

2.2. MÉTHODES DE CONCEPTION CENTRÉES UTILISATEUR

Dans un cycle de développement (cf. §2.1), une activité de conception est soutenue par des méthodes qui aident à sa réalisation. Les méthodes de support à la conception ont des objectifs divers, interviennent à des étapes différentes du cycle de vie, et participent ainsi au bon déroulement de la conception pour fournir une application finale acceptable. Parmi les attributs qui rendent une application acceptable (cf. Figure 10), l'utilisabilité concerne l'adéquation de l'application avec les besoins de l'utilisateur, son environnement et ses compétences. En positionnant l'utilisateur final et ses besoins au centre des réflexions de conception, les méthodes de cette section fournissent un support pour que l'application finale soit utilisable.

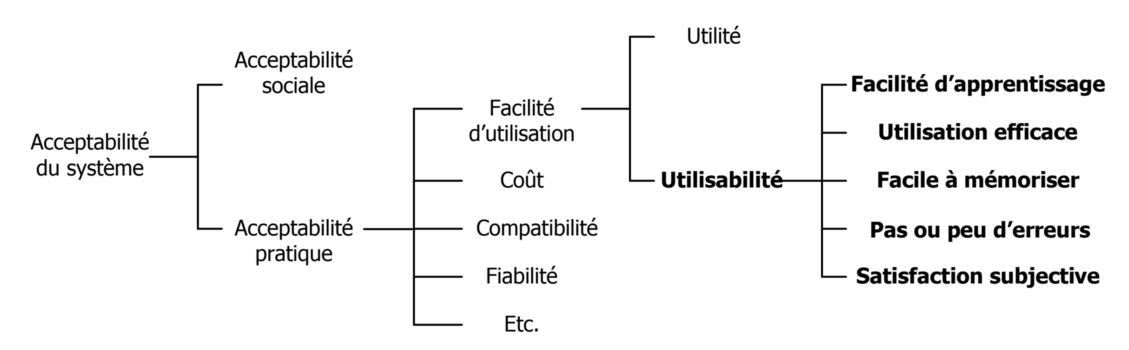


Figure 10. Variables pour l'évaluation de l'utilisabilité (issu de (Nielsen 1993))

[1] http://www.maine.gov/informe/ (dernier accès : 25 sept. 08)
[2] http://www.maine.gov/informe/state/apps/process.htm (dernier accès : 25 sept. 08)
[3] http://www.ezgov.com/ (dernier accès : 25 sept. 08)

2.2.1. Observation des utilisateurs

Un utilisateur est généralement inapte à la description précise d'un de ses comportements ou d'une de ses activités quotidiennes (Pinker 1999) : il tient pour évident ce qui ne l'est pas, et une grande partie des gestes et des raisonnements servant à atteindre un but sont inconscients. Ainsi, pour connaître son comportement, il est important de s'immerger dans son environnement d'usage. L'observation immersive permet de mieux connaître les besoins et contraintes de l'utilisateur, en appréhendant son environnement : coopérations et oppositions informelles avec d'autres personnes utilisatrices ou non de l'application, contraintes et opportunités spatiales ou sonores, interruptions. L'évaluateur doit observer le sujet comme un apprenti observe son maître d'œuvre (Beyer et Holtzblatt 1997). Encouragé par ce regard attentif et respectueux, le sujet observé fait alors preuve de lisibilité dans ses comportements et de pédagogie, en verbalisant si nécessaire. Les mêmes auteurs recommandent de 1) mener l'étude et les entretiens sur le lieu d'usage, l'utilisateur étant entouré de ses artefacts habituels, 2) établir une ambiance de partenariat où l'utilisateur coopère à l'étude, 3) lire entre les lignes des propos de l'utilisateur sans extrapoler ni interpréter, et 4) diriger les entretiens comme une conversation cordiale, avec naturel et spontanéité.

Toutefois, une observation naturaliste doit se dérouler avec la conscience qu'un biais est introduit dans le comportement d'une personne quand elle sait qu'elle est observée. L'effet Hawthorne décrit ainsi le cas où un sujet, se sentant valorisé par l'observation, manifeste une plus grande motivation dans le cadre de l'évaluation que d'ordinaire (Olson et coll. 2004). L'usage de l'enregistrement vidéo en continu peut aider à contourner ce biais dans la mesure où au bout de quelques jours, la caméra est oubliée et le naturel des comportements revient. Cela requiert toutefois une demande d'autorisation à tous les interlocuteurs, même ponctuels, de la personne observée, et qui apparaîtront donc sur les enregistrements. Les conditions nécessaires au respect de la vie privée et les coûts financiers (mise en place des caméras, maintenance, tri et analyse des enregistrements, rapport et interprétation des résultats) sont contraignants voire rédhibitoires. Des méthodes moins coûteuses peuvent toutefois être mises en place. Pour obtenir un tracé du comportement, au-delà de la période de présence de l'observateur, il peut par exemple être demandé à l'utilisateur de tenir un journal de bord. Il sera notamment intéressant, si l'utilisateur accepte de se plier à cette discipline journalière, de confronter les journaux de bord entre eux.

2.2.2. Conception participative

La conception participative prend racine dans les années 1970, au sein de projets scandinaves visant à développer des systèmes avec la participation de leurs utilisateurs (Floyd et coll. 1989). A partir d'une tâche de conception, des discussions et des activités de conception coopératives sont organisées en alternance avec des tâches individuelles parallélisées. Lors des séances de conception coopératives, le rôle social de chacun est effacé au profit de ses compétences et de ses connaissances. L'utilisateur final de l'application prend également part à ce type de conception. Il est en effet considéré comme capable d'exposer ses besoins et ses contraintes d'usage au reste de l'équipe de conception, et cette participation s'est avérée efficace dans de nombreux projets (Clement et Besselaar 1993). Une fois la confiance établie, l'utilisateur final ira jusqu'à sortir du cadre officiel de ses tâches quotidiennes, pour décrire ses tâches de façon plus officieuse. La satisfaction lors de l'utilisation après déploiement viendra ensuite non seulement du fait que l'application est utile et utilisable, mais aussi du fait que l'utilisateur a pris part à sa création. Les éventuelles modifications post-déploiement se feront également avec souplesse : le rapport d'incident sera efficace puisque l'utilisateur saura de quelle manière décrire l'incident à l'équipe technique pour qu'elle puisse la résoudre efficacement. Si les avantages de l'intégration d'utilisateurs finaux agents administratifs semblent évidents, l'utilisateur final citoyen est quant à lui plus difficile à convoquer, pour des raisons de distance d'abord (p.ex. pour une application d'e-Gouvernement à un niveau national voire international), et pour des raisons de représentativité ensuite. Pour une application donnée, le citoyen type est difficile

à identifier, et quand bien même il serait identifié, l'intégrer dans une équipe de conception serait insuffisant car il exclurait les autres cas particuliers d'utilisateurs.

L'animation de groupes de conception participative et la prise de décision sont des activités très ardues. Tous les intervenants doivent être considérés, chacun avec les informations dont il dispose, son jargon et son point de vue. Jean Caelen (Caelen 2004) préconise l'utilisation d'outils d'enregistrement des moments de conception coopérative, et de méthodes de tracé des décisions de conception. Ces dernières, connues sous le nom de *Design Rationale*[1] (Gruber et Russell 1990) offrent un moyen de conserver une trace du raisonnement qui a guidé les choix de conception d'un système. Les critères ayant guidé les choix sont listés, pondérés et argumentés pour qu'une fois confrontés les uns aux autres, la décision à prendre s'appuie sur eux de façon cohérente et exhaustive. Le tracé des décisions de conception d'un système peut également servir de documentation pour les projets de conception ultérieurs. Des notations graphiques accompagnées d'outils d'édition existent pour rendre cette activité de traçage plus utilisable (Lacaze 2005).

2.2.3. Prototypage basse fidélité

Le prototypage basse fidélité permet de spécifier l'interface d'une application de façon rapide mais suffisamment réaliste pour collecter des remarques de la part des utilisateurs finaux aussi bien que des membres de l'équipe de conception. Le prototype est conçu avec du papier, avec un éditeur de diaporama (type Microsoft PowerPoint ou OpenOffice Impress) ou de diagrammes (type Microsoft Visio). Une enquête a été menée en juin 2002 (Olsen 2002) sur les usages du prototypage basse fidélité dans la conception d'interfaces Web. La communication permise par le prototype réalisé, tant avec le client qu'avec l'équipe de conception, vient en tête des critères de choix d'un outil de support au prototypage. La facilité avec laquelle un prototype peut être modifié est également un critère important, vis-à-vis duquel les outils de programmation HTML et d'édition de diagrammes échouent. Ils restent pourtant les plus populaires dans le support à cette activité.

Un prototype basse fidélité permet de présenter à l'équipe de conception un projet d'interface ou de mener des tests d'utilisabilité. Les interactions de l'utilisateur avec l'application sont alors simulées par le meneur du test. Ces tests peuvent être de nature exploratoire (l'utilisateur navigue dans l'application au hasard, en faisant des commentaires à voix haute) ou dirigée par des scénarios (l'utilisateur doit accomplir une tâche grâce au prototype). L'utilisateur verbalise ses actions (p.ex. *Je clique sur le bouton « Me connecter »*) et le meneur du test modifie le prototype en réaction à cette action (p.ex. le meneur ôte la partie centrale du prototype et dépose à la place le morceau de papier correspondant à la page « Me connecter »). Ce type de manipulation requiert de la part du meneur une vive dextérité et une excellente connaissance des spécifications de l'interface. Dans le cas de manipulations trop lentes ou trop hésitantes, et même s'il est conscient du protocole, l'utilisateur risque de perdre le fil de ses actions et cela risque d'éloigner le déroulement du scénario de test d'un déroulement réel.

2.2.4. *Personas*

Grâce au principe des *Personas* (Cooper 1999), l'équipe de conception projette l'utilisation de l'application sur un ensemble d'utilisateurs fictifs. Cette projection est explicite, décrite dans une série de fiches descriptives de ces utilisateurs fictifs qui sera diffusée à l'ensemble de l'équipe de conception. Comme tout utilisateur, une *Persona* présente des caractéristiques standards et d'autres moins attendues. Une *Persona* aborde l'application avec ses propres compétences, elle souhaite atteindre des objectifs concrets avec l'application (p.ex. « Je veux signaler mon changement d'adresse), et des buts et objectifs plus abstraits, qui concernent l'application concernée entre autres domaines (p.ex. « je cherche à gagner du temps », « j'évite de stocker des papiers »). Un exemple de fiche est proposé dans la Figure 11, pour la conception d'une application permettant de signaler un changement d'adresse.

[1] Conception Rationalisée

Prénom Matthieu
Age 36 ans
Nationalité Française
Ville Marseille (a déménagé de Lille pour cause de mutation il y a 2 mois)
Situation familiale Vie maritale avec Leila (37 ans, professeur de Latin et Grec dans le secondaire), pas d'enfant
Formation Bac littéraire (1991), DEUG d'anglais (1994), DUT information communication option métier du livre et patrimoine (1998)
Profession Responsable du rayon « Livres tourisme » à la FNAC



Matthieu
« Le Blogger »

Niveau de compétence informatique et Web Utilisation professionnelle quotidienne d'outils de bureautique, mails et Web. Utilisation personnelle quotidienne des mails et du Web (blog de critique musicale, téléchargement de musique, consultation de ses comptes bancaires = environ 15h de surf sur Internet par semaine hors travail). Connexion ADSL depuis le domicile. Possède un PC portable personnel.
Bêtes noires techniques Les problèmes de configuration de sa connexion à Internet, les virus, la confidentialité des données transmises (quoiqu'il fasse quelques achats en ligne).
Motivation pour l'utilisation des hautes technologies Acquérir de l'information au hasard de la navigation ou bien directement en réponse à une question précise (actualités, sport, musique, culture générale), Rencontrer des gens, Gagner du temps.
Motivation pour l'utilisation de l'e-Gouvernement Gagner du temps, Recevoir des alertes email pour ne pas oublier les dates limites, Centraliser mes documents administratifs pour ne pas les égarer.
En favoris Google, Netvibes avec agrégation de flux RSS pour la lecture des blogs, Typepad pour l'édition du blog, site du Crédit Lyonnais, L'équipe, Libération, Deezer pour la musique...

Objectifs personnels et professionnels Changer de voiture / Devenir chroniqueur musique dans la presse / S'installer en Islande
Citations « L'idée de devenir célèbre grâce à mon blog m'a déjà traversé l'esprit… Ou au moins de faire commerce des billets que j'y poste : ils sont très appréciés mais mes lecteurs ne pèsent pas toujours l'énergie que ça me coûte. »
« Quand j'ai un problème technique ? Si Leila ne sait pas comment le résoudre non plus, j'appelle un collègue à l'aide !! On a eu un virus sur le PC de la maison, et sans Franck, je pense que j'aurais carrément acheté un nouvel ordinateur… ! Le plus incroyable, c'est que je tiens le rôle de Franck avec mes parents ! Ils se sont mis à l'informatique il y a un an, pour pouvoir envoyer des mails à mon frère qui vit à Prague avec sa femme et ses filles. Je joue souvent les conseillers techniques avec eux ! »

Figure 11. *Persona* « Matthieu » d'une application centralisant le signalement d'un changement d'adresse

Une interview de la *Persona* peut être jointe également, avec par exemple la description par l'utilisateur fictif d'une de ses journées type, ou encore le questionnement en profondeur de ce qu'il attend de ses interactions avec les nouvelles technologies. L'objectif d'une telle description des utilisateurs finaux est de personnifier l'utilisateur final qui, même fictif, apparaît ici incarné, ce qui a pour effet de rendre le concepteur plus sensible à ses intérêts et à sa satisfaction après le déploiement de l'application. La communication au sein de l'équipe de conception est améliorée également : pour les choix de conception de l'interface, un argument peut être défendu ou contré en invoquant l'intérêt d'une *Persona* que l'ensemble des interlocuteurs connaît.

Une fructueuse expérience d'utilisation de *Personas* est rapportée dans (Pruitt et Grudin 2003). La communication autour d'elles est intensive, et se fait au travers de multiples supports : posters, emails et site Web personnels des *Personas*, etc. Les *Personas* sont utilisées non seulement comme support à la discussion, mais aussi comme support à la documentation de conception. Par exemple, pour définir des priorités dans le développement de fonctionnalités, l'intérêt que chaque *Persona* trouverait dans cette fonctionnalité est évalué. Une moyenne pondérée (par leur représentativité) des intérêts des *Personas* permet l'établissement de priorités dans le développement des fonctionnalités.

2.2.5. Groupes de discussion[1]

L'organisation d'un groupe de discussion aide à mieux connaître les utilisateurs finaux de l'application, leurs différents besoins et contraintes, mais surtout leurs opinions divergentes et

[1] En Anglais : *focus groups*

les points de consensus entre leurs attentes. Six à dix-huit personnes en moyenne sont réunies : utilisateurs finaux connus, utilisateurs potentiels et éventuellement des décideurs ou des membres de l'équipe de conception. Un équilibre doit être trouvé dans le choix des participants entre homogénéité (pour plus d'aisance dans la prise de parole et pour favoriser une ambiance amicale (Morgan 1997)) et hétérogénéité (pour forcer les participants à argumenter leurs opinions, voir émerger les points de divergence et les points de consensus et ainsi obtenir une vue plus riche du public de l'application). Un modérateur oriente la discussion vers des sujets fixés par avance, portant sur le domaine d'application (p.ex. attitudes, attentes et préjugés vis-à-vis de l'e-Commerce), sur l'environnement existant en guise d'étude préliminaire au déploiement d'un nouvel artefact (p.ex. difficultés rencontrées et modes de fonctionnement dans un secrétariat de mairie), ou encore sur tout ou partie de l'application confrontée aux avis des participants (p.ex. essais de charte graphique sur le site intranet d'une entreprise). Les modérateurs connaissent parfaitement les spécifications de l'application mais ne sont pas membres de l'équipe de conception. Les participants peuvent être détendus et sincères : leur avis n'aura pas d'impact direct sur leurs interlocuteurs.

Le rôle de modérateur est déterminant pour la qualité des résultats de l'expérience, puisqu'il s'agit tout à la fois de pousser les participants dans leurs raisonnements, de reformuler leurs propos sans les interpréter, de faire régner une ambiance amicale et détendue sans jugement individuel, tout en conservant une parfaite impartialité. La difficulté tient également à la sélection d'un groupe de discussion qui soit représentatif ce qui aura nécessité une étude démographique approfondie du public de l'application en amont. Les réunir constitue un nouvel obstacle qui peut être contourné grâce aux outils de groupes de discussion en ligne, tels que ceux fournis par la société GreenField[1] : quoique la discussion y soit moins spontanée, ce genre d'outils permet l'accès à des utilisateurs distants, la réduction du coût de l'opération, et un archivage de la discussion par nature, ce qui réduit les possibilités de mauvaise interprétation des propos.

2.2.6. Conception centrée utilisateur pour l'e-Gouvernement

Les utilisateurs d'une application d'e-Procuration sont des agents administratifs et des administrés. Une même application d'e-Procuration peut cibler ces deux catégories d'utilisateurs, par exemple pour une application permettant la saisie puis le traitement de données venant d'un administré. Chacun des rôles pouvant accéder à l'application disposera alors d'un moyen de s'identifier pour profiter des services qui lui sont proposés. Ces deux catégories d'utilisateurs d'applications d'e-Procuration peuvent ainsi être considérées :

- Les **agents administratifs** qui utilisent une application d'e-Procuration de manière quotidienne, parfois en remplacement d'une procédure papier équivalente. L'application doit utiliser le même vocabulaire qu'eux, et s'attacher à ce que leurs actions et leur environnement habituels soient compatibles avec l'application d'e-Procuration considérée. Des options de personnalisation de l'interface ou des raccourcis peuvent par exemple être mis en place pour une utilisation experte ;

- Les **administrés** qui constituent un public très large, identifiable seulement par les tâches qu'ils sont susceptibles de réaliser avec le support de l'application d'e-Procuration. En dehors de cet objectif, leurs particularités sont inconnues : compétences en informatique, familiarité avec le vocabulaire administratif, dispositif de navigation utilisé, déficiences physiques. Pourtant, pour tous, il est primordial d'atteindre l'objectif fixé et l'application doit en fournir les moyens.

Ces deux catégories doivent être appréhendées différemment lors de la conception d'une application d'e-Procuration centrée utilisateur. Par exemple, les applications à destination d'agents administratifs peuvent être améliorées par les résultats d'études ethnographiques sur la gestion et les conséquences d'interruptions dans une activité de bureau (Martin et coll. 2002).

[1] http://www.greenfieldgroup.com/ (dernier accès : 25 sept. 08)

L'interface proposée doit être compatible avec les patrons comportementaux observés avant l'intégration de l'application.

Pour ce qui concerne les administrés, des études s'attachent à établir un mode d'interaction riche, spontané et ouvert, en particulier pour les applications d'e-Gouvernement informationnelles. Ces recherches ciblent notamment les utilisateurs administrés « immigrants numériques » (Barsalo 2004) pour qui la navigation hypertexte classique n'est pas une habitude. Citons l'exemple d'une application pour une personne âgée demandeuse d'emploi (Pearce et coll. 2006) : une interface riche lui permet d'évaluer dynamiquement, selon des critères qu'elle fait évoluer, son aptitude à retrouver un emploi, en termes de temps de recherche estimé. Les concepteurs font alors le choix de l'expérience utilisateur, à combiner à celui de l'accessibilité de l'application, éventuellement par le truchement d'une interface simplifiée, disponible de façon alternative à l'interface riche.

La **participation** de l'utilisateur final au processus de conception est, lorsqu'un climat de confiance a été établi, propice à la définition et à la prise en compte d'activités officieuses, pour une meilleure utilisabilité. Par exemple, un agent administratif pourra intervenir lorsqu'un horodatage automatique est proposé, en opposant le fait que, souvent, la date indiquée sur les dossiers de traitement n'est pas la date du jour, mais une date de convenance. De plus, en participant à la conception de son futur outil de travail, l'agent administratif devient partie prenante des évolutions de son domaine d'activité : loin de se sentir menacé par ces évolutions, sa motivation augmente avec son implication. L'utilisateur final citoyen est quant à lui plus difficile à convoquer, pour des raisons de distance d'abord (cas d'applications d'e-Gouvernement à un niveau national voire international), et pour des raisons de représentativité ensuite. Les cas de conception participative en e-Gouvernement évoquent plutôt la présence de représentants de citoyens (Oostveen et Van den Besselaar 2004).

Dans le cadre de l'e-Gouvernement, plusieurs guides de recommandations suggèrent l'utilisation de **prototypages basse fidélité**. L'état du Michigan aux États-Unis d'Amérique, préconise ainsi un cycle de développement en spirale (cf. §2.1.3), avec un prototypage papier pour les premières itérations (Tombelli 2003). Il s'agit là d'un bon compromis qui tire profit de la liberté créative autorisée par le prototypage papier, tout en passant dès que possible à des prototypes plus réalistes qui permettront de discuter de l'application et de la faire évoluer de façon raisonnable sur le plan technique.

La méthode des ***Personas*** est utilisée pour la conception d'applications d'e-Gouvernement dans l'entreprise SmalS dont nous avons pu observer les activités de conception (cf. chapitre 3) : en réunion, nous avons pu constater l'emploi quasi systématique des prénoms des *Personas* au lieu du terme abstrait d'*utilisateur*. Dans le cadre de l'e-Procuration, le public cible est très large, et bien que la tâche utilisateur d'une application fonctionnelle soit facile à cerner, l'emploi d'un « utilisateur moyen » est très peu pertinent pour une conception d'interface efficace. Par exemple, quel âge a l'utilisateur moyen d'une application de déclaration de revenus ? S'il est décidé que, en moyenne, l'utilisateur a 40 ans, on oublie le cas fréquent de l'utilisateur de 20 ans qui fait sa première déclaration d'impôts, ou celui de l'utilisateur de 70 ans qui peut être moins habitué aux interactions sur le Web. Chaque citoyen est un cas particulier et le principe des *Personas* aide le concepteur à se représenter les utilisateurs comme des personnes réelles derrière leur « ensemble de caractéristiques ».

La mise en place d'un **groupe de discussion** pour une application d'e-Gouvernement (Axelsson et Melin 2007) a révélé que, en plus de la synthèse des débats fait par les modérateurs, une synthèse devrait être demandée à chaque participant. Multiplier les points de vue sur la discussion elle-même permettrait de limiter la subjectivité de l'interprétation de tels débats.

2.3. MÉTHODES DE SPÉCIFICATION D'UNE APPLICATION INTERACTIVE

Les méthodes de conception centrées utilisateur permettent de cerner les besoins et le contexte de l'utilisation de l'application (cf. §2.2). Ce travail d'étude de l'utilisateur alimente la réflexion de conception de l'application : besoins fonctionnels, informationnels, mais aussi besoins en termes d'interface. Cette section liste les méthodes de spécification existantes pour une application interactive selon leur point de vue sur l'application, c'est-à-dire la partie de l'application qu'elles s'attachent à décrire en particulier. Puis, les méthodes de spécification spécifiques au Web sont étudiées.

2.3.1. Méthodes de spécification par la modélisation

Un modèle est un système représentant les structures essentielles d'une réalité et capable à son niveau d'en expliquer ou d'en reproduire dynamiquement le fonctionnement (Birou 1966). La modélisation d'une application interactive permet d'étudier une partie de l'application, selon un angle particulier. Le langage de modélisation, s'il est formel, permet une représentation sans ambiguïté de cet angle de vue sur l'application, pouvant être discutée et transmise au sein de l'équipe de conception, par exemple de l'analyste au développeur. Sans donner une liste exhaustive de ces méthodes de modélisation, cette section fournit des références vers des courants de modélisation significatifs.

Tâche utilisateur

Une tâche utilisateur peut être définie comme un but de l'utilisateur, associé à un ensemble de tâches et d'actions ordonnées de sorte que le but soit satisfait, à leur exécution dans un contexte particulier (p.ex. avec l'aide d'une application interactive) (Storrs 1995). La modélisation de la tâche s'applique donc à décrire l'application d'un point de vue à la fois fonctionnel et interactif. Elle servira de guide de conception ou de repère pour l'évaluation : ce découpage du but en tâches et sous tâches sert à s'assurer que l'application permet à l'utilisateur d'atteindre ses objectifs par des moyens qui sont en accord avec son contexte d'utilisation et ses aptitudes. La modélisation de tâche a vu le jour dans un contexte d'observation du travail de l'utilisateur et d'adaptation de l'outil de travail aux aptitudes de l'utilisateur. L'ergonomie et la psychologie cognitive ont ainsi posé les bases de la discipline. Voici plusieurs formalismes de modélisation de tâches significatifs dans le domaine de l'Interaction Homme Machine :

- **GOMS** *Goals Operators Methods Selectors*[1] (Card, Newell, et Moran 1983) décompose une tâche en sous tâches jusqu'à ce que chaque sous tâche puisse être évaluée en termes de temps d'exécution. Ce temps d'exécution tient compte de la réflexion et du temps d'action : il est établi d'après un modèle cognitif de l'utilisateur. Les applications peuvent ainsi être évaluées de façon quantitative, ce qui s'avère surtout pertinent pour comparer plusieurs interfaces.
- **MAD** Méthode Analytique de Description des tâches (Pierret-Golbreich, Delouis, et Scapin 1989) décrit les tâches en termes de pré et post conditions, d'états initial et final de l'environnement de la tâche, de priorités entre tâches et de possibilités d'interruptions. Les objets du domaine font ainsi partie de la description de la tâche, en particulier dans la dernière version du formalisme intitulé N-MDA (Noyau du Modèle de Description de l'Activité) (Lucquiaud 2005). Cette dernière s'attache en effet à décrire les objets de l'environnement de la tâche de façon cohérente et non ambiguë.
- **TKS** *Task Knowledge Support*[2] (Peter Johnson 1992) est une méthode historique de modélisation de tâche. Elle décrit les connaissances qui interviennent dans la planification et la réalisation d'une tâche, c'est-à-dire les objets de l'environnement aussi

[1] Buts, Opérateurs, Méthodes, Sélecteurs

bien que des informations stockées dans la mémoire de l'utilisateur. Des travaux de génération automatique d'interface ont été réalisés pour le support d'un prototypage basé sur la modélisation de la tâche (Peter Johnson et coll. 1993). L'interface est générée depuis un modèle TKS couplé à un modèle de l'utilisateur. Les travaux plus récents concernent la gestion de plusieurs tâches par un même utilisateur et les tâches coopératives (Wild, Peter Johnson, et Hilary Johnson 2004).

- **DIANE+** (Tarby et Barthet 1996) fournit un lien vers un modèle de données OPAC *natural Object PAC*[1] (Tarby 1993), dérivé du modèle PAC *Presentation, Abstraction, and Control* (Coutaz 1988), un objet naturel étant un objet ayant du sens pour l'utilisateur. Le modèle OPAC est responsable de la gestion routinière des données et les représente de façon complète, tandis que la tâche ne fait référence à une donnée qu'au travers de son implication dans la réalisation d'une tâche.
- **CTT** *Concur Task Trees*[2] (Paterno, Santoro, et Tahmassebi 1998) décompose hiérarchiquement une tâche individuelle ou coopérative et inclut les tâches du système aussi bien que les tâches de l'utilisateur et les tâches réalisées en interaction entre les deux. L'inclusion du système dans cette représentation de la tâche a pour but de permettre la génération automatique de l'interface, ou de vérifier la cohérence du modèle de tâche par rapport au modèle de l'interaction (Navarre et coll. 2001). CTT est le formalisme le plus référencé dans le domaine de l'IHM. Un exemple est présenté dans la Figure 12.

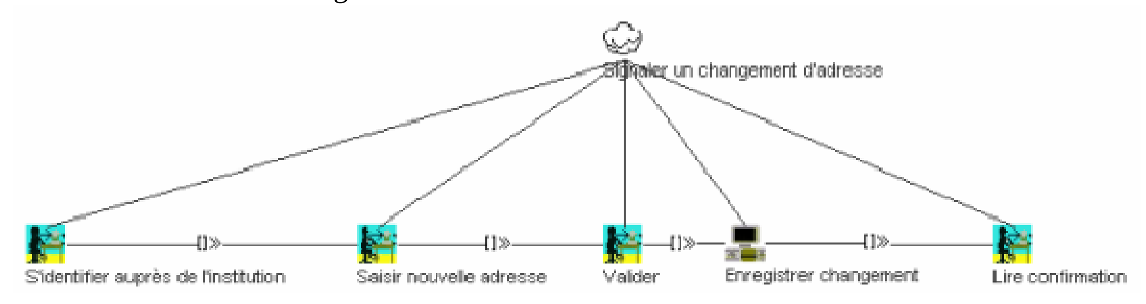


Figure 12. Modèle CTT de la tâche « Signaler un changement d'adresse »

- **GTA** *Groupware Task Analysis*[3] (Van der Veer, Martijn Van Welie, et Chisalita 2002) décompose les tâches coopératives de façon hiérarchique et inclut la description de l'environnement de la tâche, composé d'événements, d'objets, de rôles et d'agents. Il s'agit d'un cadre de modélisation à mi chemin entre la modélisation de tâche et celle de la procédure. Dans le cas d'une tâche coopérative réalisée avec le support d'une application interactive la description de la tâche s'approche de la description de la procédure, quoique ces deux types de modélisation soient issus de domaines éloignés.

Procédure

Une procédure est une séquence d'activités réalisées de manière coordonnée par plusieurs acteurs humains grâce à des ressources (documents, données, logiciels auxiliaires) afin d'atteindre un but partagé par les acteurs : transformer l'environnement dans lequel ils évoluent (le contenu d'une ressource, la structure d'une ressource, la structure de l'environnement), ou produire une ressource supplémentaire. Poussés par la complexité croissante des procédures supportées par des applications Web, des travaux ont été menés sur l'intégration d'un modèle de procédure au processus de conception Web (Koch et coll. 2003 ; Barna, Frasincar, et Houben 2006 ; Distante et coll. 2007). Plusieurs points de vue coexistent pour la modélisation d'une procédure : elle peut être vue comme une séquence d'activités utilisant des données en entrée et

[1] Objet naturel PAC (Présentation, Abstraction et Contrôle)
[2] Arbres de tâches concurrents
[3] Analyse de tâches réalisées en groupe

en produisant en sortie (p.ex. YAWL *Yet Another Workflow Language*[1] (W. M.P. Van der Aalst et Ter Hofstede 2002)), un échange de services entre agents au sein d'un environnement commun (p.ex. SophTalk (Jacobs 1992), ou encore un ensemble structuré d'informations passant au travers de stations de travail qui le modifient (p.ex. Form Flow Model (Ladd et Tsichritzis 1980)). Parmi les langages de modélisation de procédure les plus reconnus, BPMN *Business Process Modelling Notation*[2] (White 2004) se positionne en standard : la Figure 13en présente un exemple, réalisé grâce à l'outil d'édition Business Process Visual Architect 2.1[3].

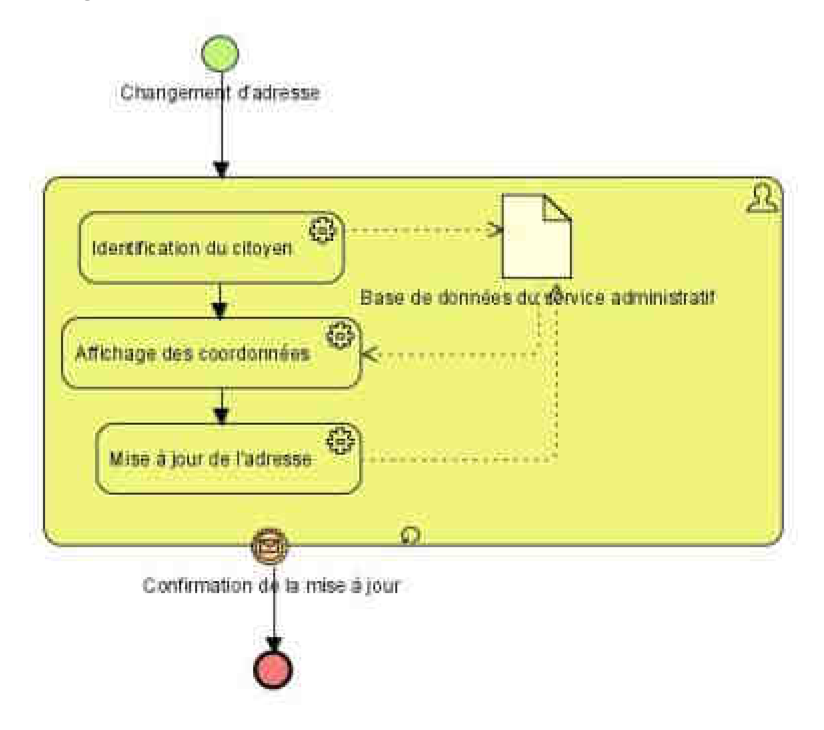


Figure 13. Modèle BPMN de la procédure « Signaler un changement d'adresse »

La modélisation de la procédure aide à la conception des interfaces d'applications fonctionnelles, dans la mesure où cette procédure globale doit être lisible sur l'interface de chaque utilisateur. Ces informations peuvent apparaître en début de procédure ou comme guidage, pour expliquer à l'utilisateur le déroulement global de la procédure. Elles peuvent également renseigner le suivi d'un dossier et notamment justifier les délais d'attente entre deux activités menées par l'utilisateur (p.ex. attente de confirmation, traitement d'un document, impression d'un document). Voici quelques unes des méthodes de modélisation de procédure existantes :

- **StateCharts**[4] (Harel 1987) décrit le comportement d'un objet et les modifications de son état en réaction à des événements qui se produisent. Un modèle StateCharts contient un état initial, un état final, et des états intermédiaires. Des transitions apparaissent entre eux, dont l'activation est conditionnée par un événement, et des conditions de garde. L'objet modélisé peut être un environnement, et l'état final est alors un état de l'environnement satisfaisant un objectif. StateCharts peut ainsi être utilisé pour modéliser une procédure avec ses contraintes et ses objectifs.
- **Les objets coopératifs** (Rémi Bastide 1992) aident à la modélisation d'un système à plusieurs niveaux d'abstraction au travers d'un formalisme basé sur les réseaux de Petri colorés (Huber, Jensen, et Shapiro 1990). Le système est représenté comme un ensemble d'objets concurrents qui coopèrent selon un mode prédéfini, dans le but de se rendre des services les uns aux autres. Ce formalisme peut ainsi modéliser une procédure.

[1] Encore un langage de modélisation de workflow
[2] Notation de modélisation de processus d'affaires
[3] http://www.visual-paradigm.com/product/bpva/ (dernier accès : 25 sept. 08)
[4] Carte d'états

- **Les diagrammes UML** (Rumbaugh, Booch, et Ivar Jacobson 1999) de séquence et de classe permettent la modélisation de données et de leur traitement, ce qui peut aider à modéliser une procédure. Les diagrammes d'activité représentent le flot des activités au travers d'une machine à états. Les objets du domaine peuvent apparaître sur ce flot, avec ceux de leurs attributs qui se voient modifiés par l'exécution d'une activité. Les activités peuvent, de plus, être groupées selon l'agent qui les réalise, sous forme de *swimlanes*[1], ce qui offre une vue organisationnelle sur la procédure représentée.
- **APM** *Action Port Model*[2] (Carlsen 1998) est un formalisme flexible permettant la modélisation d'une procédure à plusieurs niveaux d'abstraction, ce qui rend le langage utile tout au long du processus de conception. Une activité coopérative y est représentée comme un cadre de travail dans lequel plusieurs agents sont impliqués, et où des ressources sont partagées pour réaliser des tâches.
- **YAWL** *Yet Another Workflow Language*[3] (W. M. P. Van der Aalst et Ter Hofstede 2002) est un formalisme de contrôle d'exécution de flot d'activités basé sur les réseaux de Petri (Brams 1983). Des ajouts ont été faits à la notation de réseaux de Petri pour prendre en compte les besoins en spécification de patrons de procédures (Wil Van der Aalst et coll. 2003), tout en simplifiant la notation graphique pour rendre son usage plus intuitif.
- **BPMN** *Business Process Modeling Notation*[4] (White 2004) a pour but de proposer un cadre d'échange et de communication pour les acteurs du processus de conception. BPMN se positionne en tant que standard et fournit donc des langages de modélisation en prenant garde à leur lisibilité, pour qu'ils puissent constituer un véritable moyen de communication au sein de l'équipe.

Données du domaine

Pour la conception d'applications manipulant d'importants volumes d'informations, la modélisation du domaine est nécessaire. Elle permet l'explicitation de concepts connus et manipulés par tous, pour que cette représentation soit partagée de façon cohérente par les intervenants. La Figure 14 présente un exemple de modèle de domaine Entité – Relation (Codd 1970) qui pourrait être utilisé pour la conception d'une application Web centralisant le signalement de changement d'adresse auprès de plusieurs organismes. Quelques uns des formalismes disponibles pour la modélisation de domaine sont ensuite listés.

[1] Couloirs de nage
[2] Modèle de ports d'action
[3] Encore un langage pour workflow
[4] Notation de modélisation de processus métiers

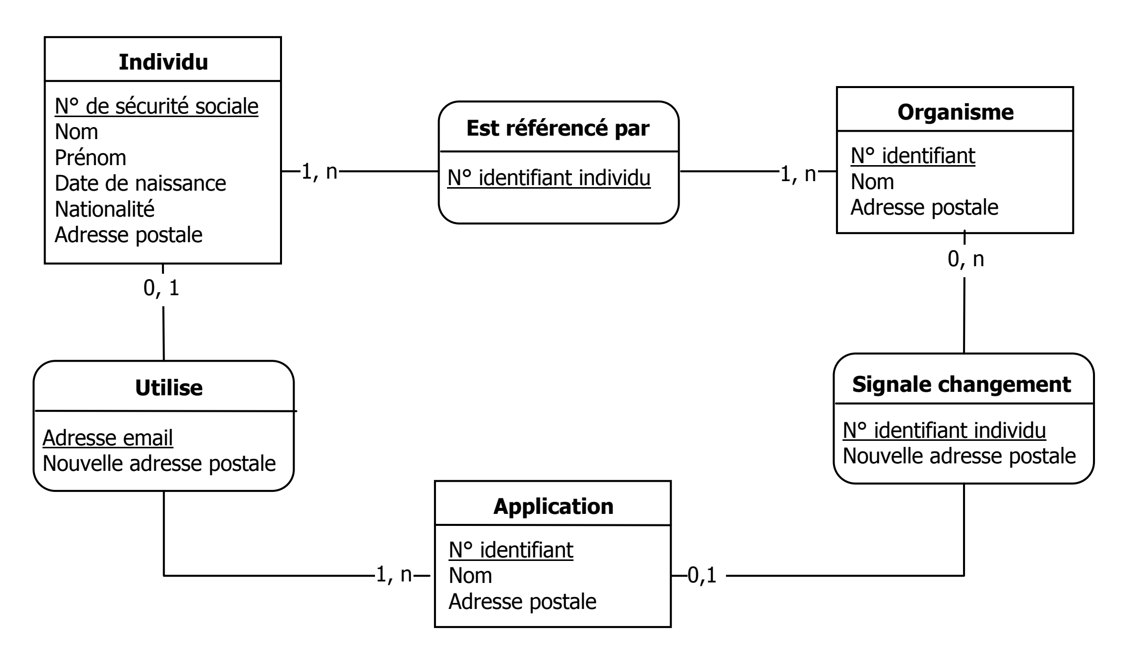


Figure 14. Modèle de domaine Entité – Relation d'une application de signalement de changement d'adresse

- **Une ontologie** décrit chaque donnée avec des métadonnées renseignant sa nature, son contexte de création et d'utilisation, ses relations aux autres données (p.ex. une page Web a été créée par une personne ; cette personne vit dans une ville ; cette ville est située dans un pays ; dans ce pays, des langues sont parlées, etc.). Ce principe permet une exploration des données très riche, proche des aspirations du Web depuis ses origines, jusqu'aux travaux très actifs concernant le Web sémantique (Berners-Lee, Hendler, et Lassila 2001). Plusieurs langages de description d'ontologies existent pour permettre leur traitement informatique, notamment les standards OWL *Ontology Web Language*[1] (W3C 2004b) et RDF *Resource Description Framework*[2] (W3C 2004a).

- **Une taxonomie** classe formellement des concepts ayant des rapports hiérarchiques complexes (p.ex. en e-Gouvernement (Suomi.fi 2007)). Les arbres de taxonomie établis permettent d'inférer des caractéristiques d'un concept sur l'autre, par des notions d'héritages.

- **Une carte heuristique**[3] permet de représenter les connexions sémantiques entre les idées, les liens hiérarchiques entre différents concepts intellectuels. C'est un pense-bête que le concepteur organise par associations d'idées, de façon très souple. Cette technique est utilisée dans la gestion de projets comme support au *brainstorming*[4], technique pouvant servir, en amont de la modélisation de domaine, pour collecter les concepts à formaliser auprès de l'équipe de conception. Toutefois, son manque de formalisme empêche une utilisation plus poussée comme support à la modélisation des données.

- **Le diagramme Entité – Relation** (Codd 1970) permet une représentation des objets du domaine telle que présentée sur la Figure 14. Une entité est un objet abstrait ou concret du domaine, caractérisé par un ensemble d'attributs. Les entités sont liées deux à deux par des relations éventuellement caractérisées par des attributs. Le nombre de ces relations est borné par des cardinalités spécifiées sur le modèle. Cette représentation du domaine est très utilisée pour spécifier des bases de données relationnelles.

[1] Langage d'ontologies pour le Web
[2] Cadre de description de ressources
[3] En Anglais : *mind map*
[4] Remue-méninges

- **Les diagrammes UML** (Rumbaugh, Booch, et Ivar Jacobson 1999) permettent une représentation statique des données (grâce au diagramme de classes), combinée à une représentation dynamique des données. Le comportement des données est alors défini en réaction à des actions ou événements (grâce au diagramme d'états) ou encore par le biais de relations temporelles entre données (grâce au diagramme de séquences). Les formalismes de la méthode UML font aujourd'hui figure de standard dans le domaine de la modélisation de domaine.

Organisation

Dans une organisation, des agents coopèrent afin d'atteindre des objectifs. Des rôles sont affectés à ces agents, sur la base de leur cursus scolaire, de formations suivies, de leur expérience, du temps qu'ils ont passé dans l'organisation. Ces rôles correspondent à des compétences, à des activités dont ils sont responsables, à des droits d'accès à des ressources informationnelles ou logicielles. Les interactions entre agents sont également décrites dans le modèle organisationnel, dans la mesure où certaines activités peuvent être déléguées de l'un à l'autre. Un agent peut ainsi recevoir temporairement, pour la durée de l'activité, des compétences étendues. La modélisation de structure organisationnelle a émergé dans le domaine des systèmes d'information. Elle est utilisée pour de nombreux systèmes informatiques, notamment ceux qui servent de support à des activités de travail coopératif. La modélisation des rôles, si elle est rendue disponible aux agents, rend l'agent conscient des compétences des agents avec lesquels il coopère, et offre une vision d'ensemble des coopérations possibles. La Figure 15 présente un exemple de modèle organisationnel, exprimé au travers d'une partie du formalisme MOT (Modèle Organisationnel des Traitements) (Tardieu, Rochfeld, et Colletti 1983).

Procédure	**Temps**		**Poste de travail**		
	Début	**Durée**	**Lieu**	**Responsable**	**Ressources**
Enregistrer un changement d'adresse	Si possible dès réception du mail du requerrant Maximum 5 jours ouvrables après	10 minutes	Poste de travail personnel	Michel Romero	Email du requerrant Accès à l'Intranet Application de gestion des affiliés

Figure 15. Modèle Organisationnel des Traitements de la procédure « Enregistrer un changement d'adresse »

Le mode de formalisation de l'organisation de MOT se place du point de vue de l'activité des agents acteurs de l'organisation. D'autres formalismes observent l'organisation du point de vue des informations accessibles à chacun, ou bien d'un point de vue dynamique d'affectation et de réaffectation des rôles en cas de besoin. Le choix des responsables de l'organisation se portera sur l'un ou l'autre de ces courants de modélisation selon les objectifs poursuivis (p.ex. meilleure souplesse de l'affectation des rôles, gestion optimale des ressources, absence de conflit des ressources disponibles, hiérarchisation des activités). Voici quelques modèles organisationnels représentatifs de ces différents courants de modélisation :

- **MOT** (Modèle Organisationnel des Traitements) est un modèle organisationnel développé dans le cadre méthodologique Merise (Tardieu, Rochfeld, et Colletti 1983). Il décrit les procédures réalisées en fonction de l'organisation et des ressources (qui, où, quand, comment). Un exemple est présenté sur la Figure 15. Les agents de l'organisation sont vus au travers des activités auxquelles ils participent et de leur façon d'y participer, ainsi qu'au travers des objectifs qu'ils aident à atteindre.
- **OSSAD** *Office Support Systems Analysis and Design*[1] (Dumas et Charbonnel 1990) est une méthodologie de modélisation du cadre méthodologique d'une organisation. Les objectifs théoriques du système sont d'abord décrits dans un modèle abstrait, en

[1] Analyse et conception de systèmes d'aide au travail de bureau

termes de services (p.ex. marketing, production). C’est le modèle descriptif qui décrit les agents, opérations et ressources qui permettent d’atteindre les objectifs du premier modèle. Un modèle prescriptif entre dans le détail des technologies utilisées.

- **TOSCA** *The Organizational Information System for CSCW* (*Computer Supported Cooperative Work*) *Applications*[1] (Prinz 1994) est une architecture orientée objet de stockage d’informations organisationnelles. Le serveur TOSCA donne ainsi des informations sur les compétences et les contraintes de l’utilisateur aussi bien que de ses collaborateurs. Ces informations, fournies d’un point de vue personnalisé à l’utilisateur, aident à établir des modes de communication et de coopération efficaces, en particulier dans le cas de coopérations à distance. Les auteurs ont pris garde à rendre ce système souple et tolérant à un grand volume d’informations, compte tenu de l’aspect évolutif d’un système coopératif.
- **Hozo** (Sunagawa et coll. 2006) est un cadre de description des rôles endossés par des agents et de leurs caractéristiques. Cette description est faite au moyen d’ontologies, pour permettre une intégration formelle du modèle organisationnel à un traitement informatique. Le contexte d’activité de l’agent est pris en compte pour la définition de son rôle : le même agent peut en effet être un citoyen dans sa vie civile et un agent administratif dans sa vie professionnelle, par exemple.
- **Mark Hoogendoorn** (Hoogendoorn et coll. 2007) propose un formalisme de description de l’organisation en se préoccupant des modifications pouvant survenir dans ce modèle, en cours d’activité, et des réorganisations qui deviennent alors nécessaires. L’étude de cas porte sur des interventions d’urgence susceptibles de survenir dans des événements sportifs de grande ampleur. Les procédures de réorganisations peuvent être internes aux agents déjà signalés ou bien consister en des extensions du modèle à des agents qui n’y apparaissaient pas. Par souci de réalisme, chaque agent est pourvu de forces motrices et de forces d’opposition aux changements de rôles qu’il subit.

Navigation

A tout moment un utilisateur doit disposer d’informations suffisantes pour savoir où il est, où il peut aller, comment faire pour aller quelque part, et comment faire pour revenir en arrière (Fleming 1998). Grâce aux informations contextuelles de navigation, l'utilisateur est capable de se situer dans le site, mais aussi d'évaluer le chemin qu'il lui reste à parcourir avant d'atteindre le but qu'il s'est fixé. La navigation au sein d’une application Web quelle qu’elle soit doit donc être traitée avec soin. Les modèles de navigation établissent l’ensemble des documents hypertextes composant l’application Web et les liens entre eux. Le modèle de navigation peut couvrir bon nombre des besoins exprimés comme requis pour la modélisation d’application Web procédurales (Winckler et coll. 2005). La Figure 16 présente un exemple de modèle de navigation StateWebCharts (Winckler et Palanque 2003) pour une application servant à signaler un changement d’adresse à plusieurs organismes. Ce modèle définit tous les scénarios de navigation possibles qu’offre l’application Web. Figurent ensuite quelques uns des formalismes de modélisation de la navigation Web disponibles actuellement, représentatifs de courants de formalismes existant dans la littérature.

[1] Le système d’informations organisationnelles pour les applications informatiques de support au travail coopératif

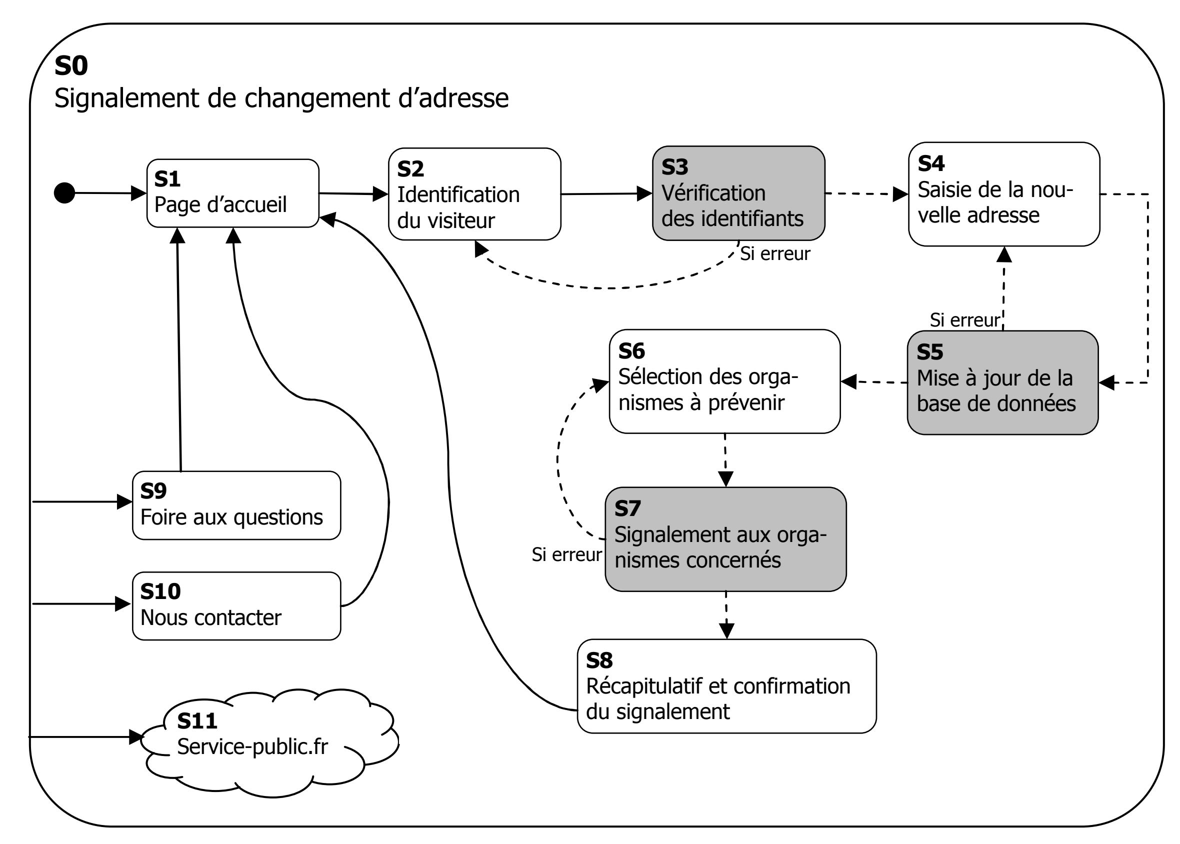


Figure 16. Modèle de navigation SWC d'une application dédiée au signalement d'un changement d'adresse

- **RMDM** *Relationship Management Data Model*[1] est un modèle qui fait partie de la méthodologie de conception Web RMM *Relationship Management Methodology*[2] (Isakowitz, Stohr, et Balasubramanian 1995). Il est inspiré des primitives de tissage de relations entre objets du diagramme Entité – Relation (Codd 1970). Il décrit tout à la fois les objets informationnels disponibles sur l'application et la structure hypertexte d'une application Web axée sur la diffusion d'information. Des patrons de navigation sont disponibles comme primitives de ce formalisme, pour représenter les différents moyens de naviguer au fil d'éléments informationnels : lien unidirectionnel, lien bidirectionnel, groupement, index, visite guidée.
- **SOHDM** *Scenario-based Object-oriented Hypermedia Design Methodology*[3] (Heeseok Lee, Choongseok Lee, et Yoo 1999) est un cadre méthodologique de conception d'applications hypermédia informationnelles. Les auteurs prétendent que les autres méthodes créent la navigation sur la base des données sans se préoccuper suffisamment du comportement de l'utilisateur. Ici, des unités de navigation sont établies, sur la base des données à manipuler, puis des scénarios sont exécutés pour observer comment et quand l'utilisateur a besoin d'accéder à ces unités. La structure hypertexte de la navigation est créée sur la base de ces observations.
- **UWE** *UML-based Web Engineering*[4] (Rumbaugh, Booch, et Ivar Jacobson 1999) modélise la navigation et la présentation d'une application Web avec des diagrammes UML de classe et de composition, après extension de leur sémantique par le principe des stéréotypes. Une page est représentée par un diagramme de composition, avec plusieurs zones symbolisant le contenu. Certains contenus sont réactifs, et mènent vers d'autres pages : ils figurent alors également sur le diagramme de classe

[1] Modèle de données pour la gestion de relations
[2] Méthodologie de gestion de relations
[3] Méthodologie de conception hypermédia orientée objet et basée sur des scénarios
[4] Ingénierie Web basée sur UML

représentant la navigation. Les liens entre pages sont en effet modélisés par des relations sur un diagramme de classe.

- **StateWebCharts** (Winckler et Palanque 2003) est un formalisme de description de la navigation dérivé de StateCharts (Harel 1987). Un modèle StateWebCharts est une machine à états hiérarchisée composée d'un nombre fini d'états et de transitions. Chaque état correspond à une page Web (appartenant à l'application ou liée depuis l'application) ou à un script exécuté côté serveur, et chaque transition symbolise un lien hypertexte entre deux pages. Un exemple de modèle StateWebCharts est présenté sur la Figure 16.
- **WSDM** *Web Semantics Design Method*[1] (Troyer, Casteleyn, et Plessers 2008) est une méthode de conception d'applications Web procédurales et sémantiques, basée sur les différents publics de l'application. Après avoir modélisé les tâches prévues pour chaque catégorie d'utilisateurs, ainsi que le modèle de données, un modèle de navigation spécifique est décrit pour chaque catégorie, sous forme de composants et de liens. Un composant est une unité de navigation regroupant information et fonctionnalités, et il est lié à d'autres composants. Un lien peut être de type structurel, sémantique, raccourci, ou bien signifier le franchissement d'une étape dans la procédure (De Troyer et Casteleyn 2003).

2.3.2. Méthodes de spécification rencontrées dans la conception Web

Plusieurs courants coexistent dans les méthodes de conception existantes, mais chacun d'eux prend soin de consacrer les premières étapes du processus de conception à la spécification de l'application. Cette section observe quelques unes des méthodes de spécification utilisées dans la conception Web, qu'elles soient publiées seules ou dans le cadre d'une méthodologie de conception complète. Quels éléments sont spécifiés ? Quelles méthodes de modélisation vues précédemment sont invoquées pour la spécification ?

JESSICA (Barta et Schranz 1998)

JESSICA est un langage de spécification des applications Web au format XML *eXtensible Markup Language*[2]. Le contenu des pages HTML y est séparé de la présentation, pour plus de flexibilité dans l'évolution de l'application Web. Chaque élément de l'application est décrit comme un objet. Les éléments multimédia, tels que la vidéo ou le son apparaissant sur une application Web sont également décrits comme des objets dans le langage JESSICA, avec des informations sur leurs caractéristiques statiques, par exemple leur nature. Des objets standard sont proposés à l'importation, ce qui facilite la réutilisation et accélère la conception si des objets efficaces ont été capitalisés par le biais de patrons. La mise en correspondance des éléments constitutifs de l'application peut permettre une modularité dans la présentation de l'application. Par exemple, il est facile de passer d'une langue à une autre sur une application spécifiée avec ce langage.

Object-Oriented Hypermedia Design Method[3] (Schwabe et Rossi 1998)

OOHDM est une méthode de conception d'applications hypermédia qui consiste en quatre étapes successives, chacune enrichissant la précédente selon un processus incrémental : modélisation conceptuelle, conception de la navigation, conception de l'interface abstraite et implémentation. Ces étapes visent à concevoir une application Web qui soit, du point de vue de la navigation et de l'interaction, adaptée aux besoins de l'utilisateur. Les auteurs prétendent que la force du Web tient dans sa nature hypermédia, c'est-à-dire dans sa faculté de lier des informations entre elles ; en d'autres termes la navigation est au centre de la méthode. Au fil de la

[1] Méthode de conception de sites sémantiques
[2] Langage de balisage extensible
[3] Méthode de conception hypermédia orientée objet

conception, les éléments suivants sont spécifiés : acteurs et tâches associées, données du domaine, scénarios d'utilisation, navigation avec prise en compte du contexte, interface. Les notations utilisées pour ces spécifications sont issues d'UML (Rumbaugh, Booch, et Ivar Jacobson 1999), et ADV *Abstract Data Views*[1] (Cowan et Lucena 1995) est utilisé pour la description abstraite de l'interface, c'est-à-dire sa description en termes d'objets avec un comportement propre, et réagissant à des événements extérieurs. Cette description orientée objet fournit l'occasion de moduler la conception de l'interface pour une réutilisation de ces modules ou une capitalisation des bons modules sous formes de patrons.

extended World Wide Web Design Technique[2] (Scharl 1999)

eW3DT est un méta modèle permettant la spécification des applications Web informationnelles. L'application peut être décrite au moyen d'objets de type information (tout contenu informationnel atomique, p.ex. une page ou un menu), navigation (tout lien hypertexte) et structure (p.ex. combinaison d'objets hiérarchiquement liés). Cette méthode peut être utilisée en conception, dans un cadre classique de processus de conception, comme support à la spécification de l'application. Un modèle eW3DT peut également être utilisé comme support à la collecte d'informations sur l'utilisation de l'application en vue d'une évaluation. Chaque objet interactif porte alors des attributs tels que le nombre de visites, ou le nombre de clics effectués sur l'objet. Les éléments spécifiables avec cette méthode sont donc les données, la navigation et la présentation.

Web Modeling Language[3] (Ceri, Fraternali, et Bongio 2000)

L'objectif de WebML est de guider la conception d'applications fonctionnelles de type B2B *Business to Business*[4] centrées utilisateurs et avec des modules qui soient réutilisables dans des projets futurs. Les concepts du modèle de la procédure servent ici à étendre le modèle de navigation pour délimiter sur ce modèle le début et la fin d'une activité, pour décomposer une activité en unités de travail, et pour définir quelles ressources sont requises pour une activité donnée. Les contraintes du déroulement de la procédure (p.ex. coordination des activités, ressource requise) sont ainsi définies par rapport au modèle de navigation de l'application. Lorsque la procédure subit des modifications, le modèle de navigation est a fortiori mis à jour, ce qui assure que l'application reflète la nouvelle définition de la procédure. Cette flexibilité est intéressante pour un domaine fortement évolutif tel que l'e-Gouvernement, dans lequel les modalités administratives sont modifiées régulièrement. Au niveau langage, WebML est basé sur le modèle Entité – Relation (Codd 1970) et sur UML (Rumbaugh, Booch, et Ivar Jacobson 1999) ; la méthode dispose d'une notation graphique ayant une syntaxe textuelle équivalente en XML.

UML-based Web Engineering[5] (Koch et coll. 2000)

UWE est un cadre de conception Web qui utilise exclusivement des formalismes issus du standard UML (Rumbaugh, Booch, et Ivar Jacobson 1999), à toutes les étapes du processus de conception classique préconisé par les auteurs : analyse des besoins, modélisation conceptuelle, modélisation de la navigation et modélisation de la présentation. UML, en tant que standard, est un formalisme largement utilisé en industrie, ce qui favorise l'acceptation de cette méthode de conception. La méthode est accompagnée de l'outil ArgoUWE (Knapp et coll. 2003) qui soutient l'édition de modèles, la vérification de consistance entre modèles conceptuels, la critique des modèles édités en fonction de recommandations ainsi que la génération semi-automatique de l'interface de l'application sur la base des modèles édités. Cette génération est rendue possible par la spécification du contenu de l'application comme un ensemble hiérarchisé de contenus

[1] Vues abstraites sur les données
[2] Technique de conception Web étendue
[3] Langage de modélisation pour le Web
[4] Interentreprises
[5] Ingénierie Web basée sur UML

typés (p.ex. image, ancre). Chaque type de contenu est associé à une présentation par défaut, qui apparaîtra sur l'interface générée. Cette fonctionnalité peut être utile pour un prototypage rapide.

Web Semantics Design Method*[1] *(Troyer, Casteleyn, et Plessers 2008)

WSDM était déjà cité dans la section précédente (cf. §2.3.1) pour le modèle de navigation que cette méthodologie de conception propose. Nous faisons ici de nouveau allusion à cette méthode dans la mesure où elle offre une particularité très pertinente pour une conception centrée sur l'utilisateur : WSDM utilise la notion de tâche utilisateur comme socle fondateur du cycle de développement. L'équipe de conception recense les utilisateurs finaux et les répertorie en fonction de l'ensemble de tâches qui leur est associée. Ces tâches sont modélisées grâce au formalisme CTT, cette activité de modélisation constituant l'étape fondatrice de la conception conceptuelle. Chaque tâche est modélisée de telle sorte qu'elle précise les données et les fonctionnalités dont elle a besoin. Des modèles complémentaires formalisant les concepts du domaine (utilisant OWL, standard de représentation des données du Web sémantique), la présentation et la navigation se greffent ensuite pour aller vers l'implémentation du site Web tout entier.

2.3.3. Méthodes de spécification pour une application d'e-Procuration

Les sections précédentes (§2.3.1 et §2.3.2) ont montré qu'une application Web peut être spécifiée sous plusieurs angles de vue, selon le moment de la conception et le propos du concepteur qui est en charge de cette spécification. Une spécification complète passe par la spécification de tous ces angles de vue, et leur mise en correspondance. La présente section s'attache à discuter des méthodes de spécification pour une application d'e-Procuration. Nous étudions d'abord les éléments d'une application d'e-Procuration dont la description est requise pour la spécification de l'application, c'est-à-dire l'importance relative des angles de vue sur une application d'e-Procuration. Les méthodes étudiées dans les sections précédentes sont ensuite comparées au vu de leur couverture des éléments listés. Des expériences de spécification pour l'e-Gouvernement issues de travaux académiques sont ensuite rapportées.

Informations nécessaires à la spécification d'une application d'e-Procuration

Une procédure administrative est un ensemble coordonné d'activités, réalisées par des agents qui en ont la compétence. Dans un domaine hiérarchisé tel que l'administration, l'allocation d'un rôle à un agent lui ouvre en effet des droits en termes d'accès à des données et de réalisation de tâches. L'allocation d'un rôle peut être temporaire (p.ex. délégation de la réalisation d'une tâche) ou définitive. La spécification d'une tâche d'e-Procuration implique donc des informations concernant le domaine, l'organisation et la procédure, comme l'illustre la Figure 17 (Pontico, Farenc, et Winckler 2006). Cette figure présente un exemple d'informations qui seraient nécessaires à la spécification d'une procédure administrative sur la base de modèles. Il s'agit d'une procédure d'inscription d'enfant à un centre aéré, dans laquelle une activité consiste à vérifier que l'enfant réside dans la ville du centre aéré. Cette activité nécessite l'accès au registre de la mairie regroupant les habitants de la ville, cette ressource n'étant autorisée qu'au directeur du centre aéré. La définition de ce simple ensemble de contraintes implique la participation de plusieurs modèles, comme il apparaît sur la figure.

[1] Méthode de conception de site Web

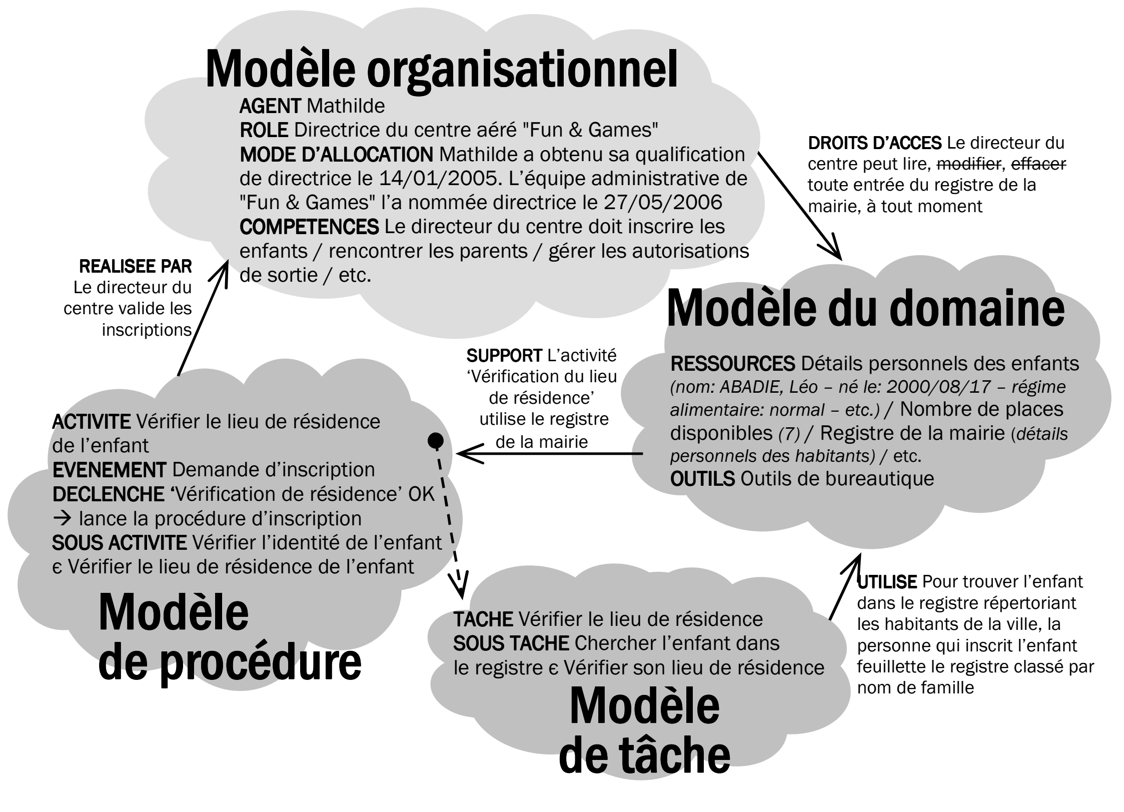


Figure 17. Exemple d'informations nécessaires à la spécification d'une procédure administrative : inscription au centre aéré « Fun & Games »

Les informations suivantes devraient apparaître pour la spécification de la procédure et des activités d'une application d'e-Procuration, toutes problématiques liées au Web mises à part.

- **Description de l'activité centrée sur l'utilisateur :** les buts pouvant être satisfaits avec l'application y sont décrits du point de vue de l'utilisateur, et décomposés en termes de tâches et de sous tâches à accomplir pour atteindre l'objectif ;
- **Intervention de plusieurs utilisateurs dans une activité :** lors de la réalisation d'une procédure administrative, au moins deux utilisateurs coopèrent, à savoir un requérant et un agent administratif en charge du traitement. Pour spécifier une application d'e-Procuration, il est donc nécessaire de pouvoir décrire la coordination de leurs activités. Ce type d'informations pourra servir à rendre compte de l'avancement d'une procédure (p.ex. « Votre demande a été pré validée. Elle est actuellement en attente de signature par le responsable du service. ») ;
- **Définition de rôles :** un rôle alloué à un agent lui confère un certain nombre de droits en termes d'accès à des ressources (informations, outils) et de réalisation de tâches. Il doit pouvoir être précisé qu'un droit réservé à un rôle peut être temporairement alloué à un agent, par délégation d'un agent dont le rôle comprend le droit en question : il s'agit du principe de délégation.
- **Utilisation d'une ressource :** la spécification doit faire figurer les données personnelles de l'utilisateur (p.ex. statut marital, date de naissance), les documents officiels (p.ex. formulaire de demande d'une fiche d'état civil, passeport) ainsi que les outils éventuellement utilisés comme support à la procédure (p.ex. scanner, logiciel permettant l'accès à la liste des véhicules déclarés comme volés). Ces ressources, les informations qu'elles contiennent et le moyen de les utiliser ont en effet un impact sur la réalisation de la procédure ;

- **Conditions temporelles :** les conditions de réalisation de tâches telles que des durées maximales ou minimales, ou des intervalles datés de temps doivent pouvoir être exprimés dans la spécification ;
- **Conditions causales :** il doit pouvoir être spécifié qu'une activité en déclenche une autre, ou bien qu'elle l'interrompt ;
- **Conditions logiques :** il doit pouvoir être spécifié que des activités sont ordonnées en séquence, en parallèle, en concurrence, alternativement, ou qu'une activité doit être exécutée en boucle ;
- **Pré et post conditions :** il doit pouvoir être spécifié qu'une condition booléenne (p.ex. le formulaire doit être signé par le requérant) doit être validée en entrée d'une activité ou en sortie ;
- **Événements temporels :** des événements déclenchés par l'expiration d'une durée maximale ou la date du jour doivent pouvoir être spécifiés ;
- **Événements de validation :** des événements porteurs de sens par rapport à l'avancement de la procédure doivent pouvoir être spécifiés (p.ex. « Après plusieurs visites sur le site Web, le citoyen a achevé sa déclaration d'impôts »).

Le Tableau 1 ci-dessous montre l'adéquation des méthodes de spécification, productrices de divers artefacts, étudiées dans la section précédente, par rapport aux besoins en spécification pour une procédure d'e-Procuration. Il apparaît qu'un point de vue sur l'application ne suffit pas à couvrir parfaitement toutes ses spécificités. GTA, modèle de tâche représentant également la procédure, les ressources et rôles, s'accorde bien aux besoins en spécification, mais manque hélas de formalisme : les éléments sont décrits en langage naturel, ce qui pose un problème de lisibilité et d'ambiguïté.

Les modèles de procédure les plus adaptés aux besoins (UML, APM et BPMN) fournissent un point de vue très complet sur les définitions liées à la procédure, mais faillissent dans la représentation de l'utilisateur individuel. StateWebCharts satisfait ou au moins touche à chacun des critères énoncés comme requis. Quoique ce point de vue manifeste de la lourdeur pour décrire des angles de vue pour lesquels il n'a pas été conçu initialement (p.ex. une tâche utilisateur), il concorde avec chacun des besoins en spécification pour une application d'e-Procuration. Parmi les méthodes spécifiques au Web, WSDM, WebML et UWE couvrent de façon très satisfaisante les besoins. Ces cadres méthodologiques nous serviront d'inspiration pour enchaîner les étapes et les formalismes à employer pour la conception d'une application d'e-Procuration.

Tableau 1. Adéquation aux besoins de l'e-Procuration des méthodes de spécification de systèmes interactifs étudiées

	TÂCHE UTILISATEUR						PROCÉDURE						DONNÉES DU DOMAINE					ORGANISATION					NAVIGATION					MÉTHODES DE SPÉCIFICATION WEB					
	GOMS	N-MDA	TKS	DIANE+	CTT	GTA	StateCharts	Objets Coopératifs	Diagr. UML (cptmt., struct.)	APM	YAWL	BPMN	Ontologie	Taxonomie	Carte heuristique	Diagramme Entité – Relation	Diagr. UML (classe, état, séq.)	MOT	OSSAD	TOSCA	Hozo	Mark Hoogendoorn	RMDM	WSDM (mod. de navigation)	SOHDM	Diagr. UML (classe, compo.)	StateWebCharts	JESSICA	OOHDM	WSDM	eW3DT	WebML	UWE
Description de l'activité centrée sur l'utilisateur	✓	✓	✓	✓	✓	✓	~	~	~	~	~	~	×	×	×	×	~	×	×	×	×	×	✓	~	✓	✓	✓	×	✓	✓	~	~	✓
Intervention de plusieurs utilisateurs dans une activité	×	×	✓	×	✓	✓	~	✓	✓	✓	✓	✓	×	×	×	×	✓	✓	✓	✓	~	✓	×	×	×	×	~	×	×	✓	×	✓	✓
Définition de rôles	×	×	~	×	~	✓	×	~	~	✓	~	✓	✓	×	×	×	×	✓	✓	✓	✓	✓	✓	✓	×	×	~	×	✓	✓	×	~	~
Utilisation d'une ressource	×	×	✓	~	~	✓	~	✓	✓	✓	✓	✓	✓	×	×	×	✓	✓	✓	✓	✓	×	✓	✓	✓	×	~	✓	✓	✓	✓	✓	✓
Conditions tempo-relles	✓	×	×	×	×	~	✓	✓	✓	✓	✓	✓	×	×	×	×	✓	✓	~	✓	×	×	×	×	×	×	✓	×	×	✓	×	✓	✓
Conditions causa-les	✓	✓	✓	✓	✓	~	✓	✓	✓	✓	✓	✓	×	×	×	×	✓	~	~	✓	×	×	×	✓	×	✓	✓	×	×	✓	×	✓	✓
Conditions logi-ques	✓	✓	~	✓	✓	~	✓	✓	✓	✓	✓	✓	×	×	×	×	✓	✓	~	✓	×	×	×	✓	×	×	✓	×	×	✓	×	✓	✓
Pré et post condi-tions	×	✓	✓	✓	✓	~	✓	✓	✓	✓	✓	✓	×	×	×	×	✓	~	✓	✓	×	×	×	✓	×	✓	✓	~	~	✓	×	✓	✓
Événements temporels	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	×	✓	✓	~	~	×	×	×	~	×	×	✓	×	×	~	×	✓	✓
Événements de validation	×	×	✓	×	×	✓	×	×	✓	✓	×	✓	×	×	×	×	×	✓	✓	✓	×	×	×	~	×	×	~	×	~	~	×	✓	~

Expériences de spécification en e-Procuration dans la littérature

De nombreux projets européens se consacrent à la modélisation de **procédure** pour l'e-Gouvernement, majoritairement avec une visée d'interopérabilité (ICTE-PAN (Loukis et Kokolakis 2003), OneStopGov[1]). Dans un état de l'art des méthodes de modélisation de procédure existantes, Olivier Glassey montre qu'elles sont insuffisantes pour l'e-Gouvernement pour ce qui concerne la représentation des données (Glassey et Chappelet 2002) et propose un moyen d'intégrer la représentation des données à la modélisation de la procédure (Glassey 2005). Le rapport d'étude sur la modélisation de procédure pour l'e-Gouvernement de David Brain met en avant les concepts nécessaires pour une modélisation pertinente (Brain, Seltsikas, et Tailor 2005) : activité (coordination, entrées et sorties, ressources requises, acteur, besoins légaux, besoins en sécurité, localisation), document (propriétaire, origine, connexion à d'autres domai-

[1] http://www.onestopgov-project.org/ (dernier accès : 25 sept. 08)

nes d'activité), environnement de travail (acteurs, unités organisationnelles, environnements subordonnés). L'utilisateur d'une application d'e-Gouvernement est ainsi indirectement défini par des activités à mener, des ressources dont il est propriétaire, et une appartenance à une unité organisationnelle. Ce rapport établit également qu'une contrainte dans la définition d'une procédure peut être légale, technique, sociale, éthique et géographique.

C'est également l'objectif de l'interopérabilité qui mène les recherches en modélisation de **données** pour l'e-Gouvernement. Les principes du Web Sémantique sont largement utilisés, ce qui augmente les relations possibles entre concepts, les inférences entre informations et ainsi le pouvoir sémantique du modèle de domaine : classification avancée des concepts manipulés par des taxonomies (Suomi.fi 2007) ou des ontologies (Efthimios Tambouris et coll. 2004 ; Wimmer 2006) ; étiquetage des concepts par des métadonnées pour établir des cartes de connaissances (Handzic 2004). Des efforts importants sont menés pour l'établissement d'un langage commun à tous les services administratifs, au niveau international, première étape pour un succès dans l'établissement de telles architectures d'interopérabilité. Plusieurs stratégies sont ainsi remarquables dans cet effort de description de services administratifs standardisée : GovML (Kavadias et Efthimios Tambouris 2003) est un langage de marquage pour la description de services administratifs tandis que (Fraser et coll. 2003) se concentre sur l'utilisation de taxonomies basées sur une ontologie des services d'e-Gouvernement. Une fois ce langage établi, il sera alors possible de mettre en place une plateforme d'interopérabilité des services administratifs, fournissant un portail *One-Stop* aux citoyens.

2.4. MÉTHODES D'ÉVALUATION DE L'UTILISABILITÉ

Les premiers processus de conception proposés par les acteurs du génie logiciel se focalisaient sur l'utilité de l'application et, si une étape d'évaluation apparaissait, elle était consacrée aux tests techniques des fonctionnalités de l'application, sans se soucier de l'interaction requise pour utiliser ces fonctionnalités. Puis l'IHM et le génie logiciel ont croisé leurs réflexions, les systèmes informatiques se sont démocratisés, certains ont été abandonnés faute d'être en adéquation avec leur public, et l'utilisateur a naturellement trouvé sa place dans l'évaluation. Dès lors, une application, en plus d'être utile, se doit d'être utilisable, et le rôle du concepteur est de s'en assurer avant le déploiement. Pour réduire les coûts des modifications, des travaux sont menés pour que l'évaluation puisse être menée le plus tôt possible dans le processus de conception, sur des artefacts intermédiaires, prémices de l'application finale.

Cette exigence d'utilisabilité vaut pour les applications interactives mais aussi pour les méthodes de conception. Les concepteurs sont des utilisateurs de méthodes, et celles-ci doivent être en adéquation avec leurs besoins et leurs aptitudes. Ce travail de thèse s'attache à établir une méthode de conception pour des applications d'e-Gouvernement utilisables. La méthode elle-même se doit d'être utilisable pour les concepteurs. Cette section a pour but, non de couvrir exhaustivement les méthodes d'évaluation de l'utilisabilité, mais d'offrir un panorama des méthodes d'évaluation qui nous serviront à nous assurer de l'utilisabilité et de l'acceptabilité de la méthode que nous proposons (cf. chapitre 5).

2.4.1. Observation de faits provoqués

Dans la section 2.2.1, les méthodes d'observation permettaient, en amont, de recueillir les besoins des utilisateurs et de prévoir les conditions idéales de déploiement d'un nouvel artefact. Nous reprenons ici les mêmes principes, mais pour le cas d'un artefact qui a déjà été inséré dans l'environnement de l'utilisateur, et qui fait l'objet d'une évaluation de son utilisabilité. Il s'agit de la méthode la plus sûre de détection de problèmes d'utilisabilité, puisque l'observation de l'activité est au centre de la méthode. Lorsque des hypothèses sont émises par avance par l'évaluateur, et que l'observation a pour but de les vérifier (p.ex. les utilisateurs réalisent leurs activités quotidiennes plus rapidement grâce au nouvel artefact), on parle d' « observation armée » dans un contexte de faits provoqués par l'évaluateur (par opposition à une observation pure et simple des faits tels qu'ils se produiraient d'eux-mêmes).

La technique du *Thinking Aloud*[1] (H. Rex Hartson et coll. 1996) est un support à l'observation de faits provoqués : il est demandé à l'utilisateur de verbaliser un maximum ses pensées, ressentis et raisonnements, afin de saisir au plus près les mécanismes de son utilisation de l'application. Cette méthode de verbalisation s'avère plus efficace, car plus naturelle, lorsque l'observateur est physiquement proche de l'utilisateur, par opposition aux cas d'observations à distance, avec enregistrement vidéo par exemple. En ce qui concerne les informations à relever lors de ces observations, JoAnn Hackos (Hackos et Redish 1998) suggère à un observateur de noter, pour une tâche donnée :

- La situation de l'utilisateur et de son environnement en début de tâche ;
- L'heure de début de la tâche ;
- Le déclencheur de la tâche ;
- Le comportement de l'utilisateur au cours de la réalisation de cette tâche, de la façon la plus objective possible ;
- Le moment où l'utilisateur considère que la tâche est terminée ;
- L'issue de la tâche : l'objectif a-t-il été atteint ? ;
- L'heure de fin de la tâche.

La mise en place d'observations où l'évaluateur est aux côtés de l'utilisateur requiert une bonne organisation mais peu de ressources matérielles. Toutefois, l'observateur doit faire preuve d'une grande expertise et présenter les qualités suivantes : capacité d'instaurer un climat de confiance (l'utilisateur doit comprendre et ressentir que c'est l'application qui est évaluée, pas lui), attention et spontanéité (l'observateur doit noter toutes les attitudes et les actions de l'utilisateur, et éventuellement engager un entretien improvisé avec l'utilisateur au sujet d'un de ces événements) et neutralité (l'observateur doit mettre de côté ses éventuels a priori sur l'utilisateur et l'application pour se concentrer sur l'observation de l'activité réelle brute, sans l'interpréter). En dépit du faible besoin en matériel, la présence requise d'un expert, pour l'observation et pour l'analyse des données collectées, rend ainsi la méthode très coûteuse.

2.4.2. Consultation de l'utilisateur

Il s'agit ici de questionner l'utilisateur sur ses sentiments vis-à-vis de l'artefact à évaluer. Cette activité est généralement menée par le biais de questionnaires, faciles à diffuser largement au format papier ou au format électronique, via un site Web dédié (p.ex. SurveyMonkey[2]) ou par courrier électronique. Des questions ouvertes ou fermées permettent de recueillir respectivement des informations plutôt qualitatives ou quantitatives. Plusieurs types de questions se côtoient encore, celles de type factuel (p.ex. informations démographiques), celles portant sur une appréciation (p.ex. préférences d'utilisation entre plusieurs artefacts) ou encore sur une attitude (p.ex. impression d'efficacité accrue ou diminuée avec l'arrivée d'un nouvel artefact). Plusieurs motivations amènent à l'utilisation de questionnaires, parmi lesquelles :

- Mesure de l'utilisabilité d'une application : plusieurs questionnaires existent dans la littérature pour le Web, tels que QUIS *Questionnaire for User Interaction Satisfaction*[3] (Harper, Slaughter, et Norman 1997) ou WAMMI *Website Analysis and MeasureMent Inventory*[4] (Kirakowski et Claridge 1998) ;
- Mesure du taux d'acceptation d'une application : le questionnaire se basera alors sur des modèles tels que UTAUT *Unified Theory of Acceptance and Use of Technology*[5] (Venkatesh et coll. 2003) qui découpent l'acceptation en satisfaction subjective,

[1] Penser à voix haute
[2] http://www.surveymonkey.com/ (dernier accès : 25 sept. 08)
[3] Questionnaire de satisfaction de l'interaction utilisateur
[4] Analyse de site Web et inventaire de mesures
[5] Théorie unifiée d'acceptation et d'utilisation d'une technologie

sensation que le travail est facilité ou rendu plus efficace par l'application, influence sociale, conditions facilitatrices à l'utilisation (p.ex. formation préalable) ou encore propension à l'utilisation de cette application ;

- Collecte d'incidents critiques (H. Rex Hartson et Castillo 1998) : l'utilisateur remplit et envoie le questionnaire dès qu'il est confronté, dans son utilisation quotidienne, à un incident avec l'application. Le questionnaire s'attachera alors à être rapide à remplir (pour encourager l'utilisateur à acquérir le réflexe de le faire dès qu'un incident survient) quoique suffisamment précis (pour que l'équipe technique puisse intervenir) : il sera en outre demandé de façon détaillée le contexte dans lequel est survenu l'incident (tâche en cours, trois derniers écrans affichés, etc.)

Les incohérences dans les réponses d'importance doivent pouvoir être détectées ; par exemple, la même question peut être posée plusieurs fois, chaque fois formulée différemment. Pour forcer le répondant à prendre position sur une question de satisfaction, ou engageant son opinion de quelque manière que ce soit, un nombre de réponses pair est recommandé (p.ex. *Pas du tout d'accord – Plutôt pas d'accord – Plutôt d'accord – Tout à fait d'accord*). Ces conseils de composition des questionnaires sont toutefois délicats à manier pour un novice. Si cette méthode, pour un coût modique, permet d'obtenir un grand volume d'informations, il faut prendre garde aux erreurs d'interprétation des questions aussi bien que des réponses. Le langage naturel est porteur d'ambiguïtés et, pour que les résultats soient pertinents et utilisables, il est bon que les questions soient écrites de façon conjointe par des techniciens, des linguistes et des experts en facteurs humains.

2.4.3. Analyse de données d'usage

Pour connaître le comportement de l'utilisateur sans se rendre auprès de lui, il est parfois possible de tracer ses interactions avec un artefact. Dans le cadre d'une application Web, chaque visite de page active le téléchargement de cette page depuis le serveur. Un ensemble d'informations pouvant servir à l'évaluation est ainsi associé à chaque téléchargement : date et heure de la requête, référence de la page demandée, référence de la page source, informations sur le poste client (système d'exploitation, navigateur, adresse IP, etc.) Des outils existent pour le traçage et la génération de graphiques facilitant l'analyse statistique des données recueillies : citons par exemple Google Analytics[1] ou WebTrends[2]. A partir de ces données, Jean-Pierre Norguet (Norguet, Limbourg, et Stijfhals 2007) propose les analyses suivantes :

- Pages d'entrée et de sortie : les entrées sur le site suite à des recherches sur des moteurs de recherche sont-elles pertinentes par rapport aux mots-clefs ? Quel est le vocabulaire choisi pour ces mots-clefs ? Les sorties du site se font-elles de façon récurrente sur une même page ? ;
- Recherches au sein du site, par moteur de recherche interne : les résultats renvoyés sont-ils pertinents ? Quels sont les mots clefs utilisés ? En cas de recherche récurrente, il est envisageable de remettre en cause l'accès à la page correspondante : le lien n'est-il pas assez visible, ou pas assez direct ?
- Différenciation du comportement d'un utilisateur qui revient après avoir déjà visité l'application (utilisation de cookies pour tracer un utilisateur unique) ;
- Observation de séquences de pages laissant à penser que l'utilisateur a atteint un but (p.ex. remplissage de formulaires successifs).

Les évaluateurs seront cependant attentifs au fait que plusieurs mécanismes faussent le contenu des données d'usage. Le premier mécanisme est le principe de cache qui masque certains affichages de page et donc fausse le nombre de visites tracées côté serveur. En effet, pour chaque page téléchargée, une copie est enregistrée en local dans le cache. Lors de la deuxième

[1] http://www.google.com/analytics/fr-FR/ (dernier accès : 25 sept. 08)
[2] http://www.webtrends.com/ (dernier accès : 25 sept. 08)

visite sur cette page, c'est la copie qui est affichée ; la page n'est pas téléchargée de nouveau, et ce pour accélérer la navigation. L'utilisateur aura affiché la page deux fois, un seul téléchargement aura pourtant été tracé depuis le serveur. Le deuxième mécanisme est l'utilisation de technologies telles qu'Ajax rendant caduque la correspondance entre un téléchargement et l'écran affiché à l'utilisateur, dans la mesure où la page Web exécute des scripts côté client et n'accède au serveur que de façon ponctuelle et asynchrone. Un troisième mécanisme faussant les tracés de données d'usage côté serveur est le suivant : certaines visites tracées ne correspondent pas à des utilisateurs, mais à des robots envoyés par des moteurs de recherche pour explorer les pages à référencer.

Une fois tous ces mécanismes pris en compte, il reste à évaluer les données collectées. Il s'agit d'une activité pour le moins ardue et discutable : comment décider qu'une séquence de pages visitées est suspecte ? Qu'une activité est symptomatique d'un problème d'utilisabilité ? Pour une application Web, par exemple, comment établir qu'un temps de visite sur une page est anormalement long ? L'utilisateur peut avoir mis en pause sa navigation le temps de réaliser une autre activité, ce n'est pas pour autant le signe d'un problème d'utilisabilité.

2.4.4. Confrontation de l'interface à des règles ergonomiques

Un artefact peut être évalué par des experts en facteurs humains, par le biais d'une inspection de l'interface, sur la base de la propre expertise de l'évaluateur ou bien d'un ensemble de règles ergonomiques. Pour guider les experts dans cette activité d'inspection, des recueils de règles ergonomiques sont publiés. Les critères d'utilisabilité pour le Web de Christian Bastien et Dominique Scapin (Christian Bastien et Scapin 1995) considèrent, par exemple, des problématiques de guidage, charge de travail, contrôle explicite, adaptabilité, gestion des erreurs, homogénéité et cohérence, signifiance des codes et dénominations et compatibilité. Le recueil de Gerry Gaffney (Gaffney 1998), quant à lui, liste des critères à vérifier pour s'assurer de l'utilisabilité d'une application Web et pointer d'éventuels problèmes de navigation, fonctionnalité, contrôle de l'utilisateur sur ses actions, langue, interaction, logique, prévention et correction des erreurs ou clarté visuelle.

Un autre mode d'inspection possible d'une interface se concentre sur la détection de problèmes liés à l'apprentissage, à la prise en main d'une application : le *Cognitive Walkthrough*[1] (Lewis et coll. 1990). Un évaluateur, expert en facteurs humains de préférence, explore l'artefact, qu'il en soit à un stade de prototype ou sur le point d'être déployé. Pour ce faire, il exécute des scénarios d'usage, pas à pas, s'interrogeant à chaque étape sur l'ensemble des informations et des aptitudes requises pour pouvoir passer à l'étape suivante. C'est en effet ce mode exploratoire, pas à pas, qui est le plus souvent employé par les utilisateurs pour appréhender l'utilisation d'un nouvel artefact. Cette méthode d'évaluation exploratoire a pour objectif la détection des éventuels obstacles à cet apprentissage. Pour guider l'évaluateur, des pistes pertinentes d'exploration sont suggérées dans (Wharton et coll. 1994).

Quoique l'activité réelle de l'utilisateur ne soit pas prise en compte dans ce type de méthode d'évaluation, des résultats rapides et intéressants sont obtenus à moindre coût en comparaison avec la mise en place de tests utilisateurs. La conjonction de plusieurs inspections est toutefois préférable : Jakob Nielsen prétend que les problèmes d'utilisabilité détectés par les inspections ergonomiques de cinq évaluateurs couvrent 75% des problèmes d'utilisabilité qui auraient émergé de tests utilisateurs (Nielsen 1994).

L'automatisation de cette évaluation, lorsqu'elle est possible, lève les problèmes d'interprétation et de non exhaustivité d'une évaluation réalisée manuellement. Certaines des règles présentes dans ces recueils de recommandations ergonomiques peuvent en effet être évaluées automatiquement sur le code de l'application, par des outils de vérification (Ivory 2003). Lorsque le code seul ne suffit pas à évaluer une règle (p.ex. pour les règles concernant la cohérence du contenu, difficile à mesurer automatiquement en l'état actuel des outils d'analyse tex-

[1] Exploration cognitive

tuelle), l'outil peut suggérer à l'évaluateur l'exploration manuelle d'une partie de l'application. Dans d'autres cas de règles que le code seul ne peut pas valider ou invalider, l'évaluateur doit saisir des informations, sur demande de l'outil, pour permettre l'évaluation. Parmi les outils disponibles pour l'évaluation ergonomique automatique des applications Web, DoctorHTML (Flanders 2003) permet d'analyser en ligne, pour une application Web, les liens brisés, le texte (analyse lexicale), la taille des images et des pages, et de vérifier la compatibilité du site auprès de différents navigateurs. DESTINE *Design & Evaluation STudio for INtent-based Ergonomic web sites*[1] (Mariage et coll. 2004) permet quant à lui d'évaluer des règles décrites au format GDL *Guideline Definition Language*[2] (Beirekdar, Vanderdonckt, et Noirhomme-Fraiture 2002) : ces règles spécifient des attributs requis sur des objets Web pour garantir l'utilisabilité de l'application Web. L'utilisation de ces outils requiert un stade avancé dans le développement de l'interface, ce qui repousse le moment de l'évaluation en fin de processus de conception. Des travaux explorent la vérification automatique sur des artefacts intermédiaires au cours du processus de conception, pour ainsi détecter d'éventuels problèmes d'utilisabilité plus tôt dans le processus de conception (Xiong, Farenc, et Winckler 2007).

2.4.5. Évaluation de l'utilisabilité d'applications d'e-Gouvernement

Les méthodes de conception d'e-Gouvernement n'ont pas, à notre connaissance, fait l'objet d'évaluations d'utilisabilité qui aient été publiées. Les retours d'expérience ci-après concernent donc les expériences rapportées dans la littérature sur l'évaluation de l'utilisabilité d'applications d'e-Gouvernement elles-mêmes.

Fred B. Wood (Wood et coll. 2003) propose une synthèse des méthodes d'évaluation de l'utilisabilité et des performances d'une application d'e-Gouvernement. L'intégration de plusieurs méthodes d'évaluation au fil des étapes de conception est recommandée, avec des avantages complémentaires en termes d'apport d'informations. Ces méthodes sont synthétisées dans le Tableau 2 : le moment adéquat d'utilisation d'une méthode d'évaluation y est mis en avant.

Pour ce qui concerne les retours sur des expériences concrètes d'évaluation de l'utilisabilité d'applications d'e-Gouvernement, Asbjørn Følstad (Følstad, Jørgensen, et Krogstie 2004) relève que sur seize projets de conception d'e-Gouvernement menés en Norvège, seulement trois ont mené des **tests utilisateurs**. Ce faible résultat s'explique en grande partie par le coût élevé de tests utilisateurs et la nécessité d'une expertise en facteurs humains pour l'analyse des résultats. Lili Wang (Lili Wang, Bretschneider, et Gant 2005), sur la base d'un état de l'art des méthodes d'évaluation de l'utilisabilité adaptées à l'e-Gouvernement, observe également un manque d'inclusion de l'utilisateur à la conception en e-Gouvernement. De façon générale, l'évaluation reste statique et ne prend pas en compte l'aspect comportemental de l'utilisateur.

De nombreuses **enquêtes** d'usage de l'e-Gouvernement sont publiées sur le Web. Elles portent souvent sur l'usage et la satisfaction envers l'e-Gouvernement des citoyens d'un pays donné, comme par exemple l'enquête démographique d'usage et de satisfaction de l'e-Gouvernement en Australie (AGIMO 2006). Toutefois, les enquêteurs échouent régulièrement dans la mise en évidence de facteurs concrets motivant et démotivant l'usage d'applications d'e-Gouvernement (Kunstelj, Jukic, et Vintar 2007). Suite à une enquête téléphonique menée auprès de 1028 citoyens en Slovénie, les auteurs ont établi un graphe explicitant les facteurs ayant un impact sur l'utilisation et la satisfaction vis-à-vis de l'e-Gouvernement.

[1] Studio de conception et d'évaluation de sites Web avec une volonté ergonomique
[2] Langage de définition de recommandation

Tableau 2. Critères de sélection d'une méthode d'évaluation d'application d'e-Gouvernement (issu de (Wood et coll. 2003))

MÉTHODE D'ÉVALUATION	ÉTAPE DU CYCLE DE VIE DU SITE WEB		
	DÉVELOPPEMENT	DÉPLOIEMENT	MAINTENANCE
Test d'utilisabilité Conception Web, navigation, accueil de l'utilisateur, fonctionnalités			
Inspection experte ou heuristique	✓✓		✓✓
Test en laboratoire d'utilisabilité	✓✓✓		✓✓
Retours informels sur l'utilisabilité	✓		✓
Retours de l'utilisateur Profil de l'utilisateur, informations démographiques, satisfaction, utilisation et impact			
Enquête interne, mise à disposition sur le Web		✓✓✓	✓✓✓
Enquête externe, mise à disposition sur le Web		✓	✓
Groupes de discussion	✓✓	✓✓	✓✓✓
Enquête diffusée à échelle nationale	✓	✓✓	✓✓
Retour spontané d'utilisateurs		✓✓	✓✓
Données d'usage Pages vues, nombre de visites, visiteurs uniques, recherches effectuées			
Analyse de données de log Web		✓✓✓	✓✓✓
Mesure de l'audience du site Internet	✓	✓✓	✓✓
Performance Web et Internet Temps de téléchargement d'une page, bande passante disponible, latence, perte de paquets			
	✓	✓✓	✓✓

LÉGENDE ✓✓✓ très important ✓✓ important ✓ moins important
Pas de marque : non applicable ou non important

En dépit de la difficulté d'analyse des **données d'usage** collectées lors de la navigation des utilisateurs en conditions réelles d'utilisation de l'application Web, ces données sont souvent collectées par les applications d'e-Gouvernement. Afin de savoir dans quelle direction les données sont collectées, et quels indices sont recherchés, des KPI *Key Performance Indicators*[1] sont utilisés, tels que ceux proposés par l'état de Victoria en Australie (Victoria Online 2006) : usage du site (nombre mensuel de pages visitées, de visiteurs uniques, de sessions), localisation des visiteurs (visiteurs de l'état de Victoria et autres), rapport de visiteurs déjà venus et de nouveaux visiteurs, sites d'origine des visiteurs (mots clefs saisis dans des moteurs de recherche, sites référents), nombre d'inscriptions à une lettre d'information, nombre d'abonnements à un fil RSS *Really Simple Syndication*[2], mots clefs saisis dans le moteur de recherche interne. Les données d'usage collectées peuvent être exploitées automatiquement pour l'interface de l'application, et permettre, par exemple, un accès rapide aux pages que le visiteur souhaite probablement visiter (c'est-à-dire ayant un nombre de visites important).

[1] Indicateurs clefs de performance
[2] Syndication vraiment simple. Lorsqu'un utilisateur s'abonne au fil RSS d'un site Web, il est alerté dès que des mises à jour sont effectuées sur le site.

Considérant qu'une **inspection heuristique** n'a de valeur que pour un ensemble d'heuristiques qui soit adapté au contexte de déploiement de l'application, Ana Cristina Bicharra Garcia (Bicharra Garcia, Maciel, et Bicharra Pinto 2005) propose une extension des heuristiques pour le Web de Jakob Nielsen (Nielsen 2005) pour le cas particulier de l'e-Gouvernement. Les critères à satisfaire restent les mêmes : le concepteur doit prendre garde à minimiser la charge cognitive de l'utilisateur, à ménager sa tolérance aux lenteurs du système, à offrir une large portée à l'application, pour tout type de support et pour tout type d'utilisateur, à réduire l'effort physique lors de l'utilisation et à encourager la confiance que l'utilisateur place dans l'application. Aux composants proposés par Jakob Nielsen pour satisfaire ces critères (p.ex. visibilité de l'état du système, prévention des erreurs), les auteurs ajoutent des composants requis dans le cadre de l'e-Gouvernement : accessibilité, interopérabilité, sécurité et respect de la vie privée, fiabilité de l'information, souplesse du service et transparence. Au-delà de ces heuristiques, d'autres recommandations ergonomiques peuvent alimenter l'inspection experte d'une application d'e-Gouvernement, par exemple des conseils pour la conception de formulaires électroniques (Axelsson et Ventura 2007).

2.5. MÉTHODES DE GESTION DE LA CONNAISSANCE ERGONOMIQUE

Les méthodes de gestion de la connaissance ergonomique viennent en complément des méthodes d'évaluation de l'utilisabilité (cf. §2.4), pour penser et éviter les problèmes d'utilisabilité de l'application finale de façon préventive. Ces méthodes et outils servent de support lors de la conception, en diffusant des bonnes pratiques de conception, sous forme de recommandations ou bien de canevas de solutions prêtes à l'emploi. Le concepteur, face à un contexte particulier, puise ainsi dans ces méthodes des moyens de s'assurer lui-même du bien-fondé de ses choix en termes de conception d'interface.

2.5.1. Règles ergonomiques

Les informaticiens puisent dans les ressources produites par des experts en facteurs humains pour rendre leurs interfaces ergonomiques. La présence d'un expert en ergonomie dans une équipe de conception est un atout précieux pour l'évaluation de l'utilisabilité, la direction de tests utilisateurs pertinents et l'analyse des besoins utilisateurs. Pour les étapes de conception aussi bien que d'évaluation, des recommandations ergonomiques ont été rendues exploitables (Nielsen 1990 ; Scapin 1990 ; Vanderdonckt 1994). Il existe des recommandations liées à un contexte d'interaction particulier tel que le Web (Scapin et coll. 2000 ; Nielsen 2006) ou les applications multimodales (Reeves et coll. 2004). D'autres recommandations sont dédiées à des domaines d'application tels que la télé santé (Niman et coll. 2007) ou l'e-Gouvernement (Maguire 1999 ; UK e-Government unit 2004 ; Section508.gov 2008). Dans un domaine particulier, ou dans le cadre d'une entreprise, les recommandations ergonomiques sont un moyen de définir précisément leurs exigences en matière d'utilisabilité (p.ex. « Le site Web doit être accessible aux utilisateurs équipés d'un lecteur d'écran audio ») et de fournir en même temps des moyens d'évaluer ces mêmes exigences sur des applications existantes. De même les règles ergonomiques constituent un format adéquat pour l'expression des standards de bonne conduite de conception, édités par les organismes de standardisation internationaux. A titre d'exemple, la recommandation de la Figure 18 est extraite des recommandations nationales du Royaume-Uni pour la conception d'applications d'e-Gouvernement ; la Figure 19 présente quant à elle un standard d'accessibilité issu de l'initiative du WAI (*Web Accessibility Initiative*[1]).

[1] Initiative pour l'accessibilité du Web

2.2.4 Adresses de contact

Une adresse postale complète doit être fournie, ainsi qu'une adresse email.

Les adresses personnelles doivent être évitées ; par exemple richard@hotmail.com devrait être évitée au profit de richard@e-envoy.gov.uk

Sinon, des adresses génériques peuvent être utilisées, si la stratégie de votre organisation le permet, par exemple webguidelines@e-envoy.gsi.gov.uk. Ce type d'adresse permet la portabilité des adresses email en cas de changement de personnel.

Quand des adresses email sont insérées dans une page HTML, elles doivent être affichées au format universel.

L'affichage du seul nom des personnes ne suffit pas car :

- un utilisateur novice du Web pourrait ne pas réaliser qu'il peut contacter la personne si seul son nom est affiché
- des utilisateurs peuvent avoir modifié la configuration par défaut de leur navigateur Web. Ils peuvent avoir désactivé le soulignage des liens, ou avoir configuré la couleur des liens comme identique à la couleur du texte standard.

Figure 18. Recommandation du UK e-Government unit pour l'affichage des coordonnées de contact (issue de (UK e-Government unit 2004) (UK e-Government unit 2004))

3.9 Clignotement d'écran

Un écran clignotant ou flashant peut provoquer des crises chez certains utilisateurs atteints d'épilepsie photo sensitive et les développeurs de contenu devraient éviter de faire clignoter l'écran. Les crises peuvent être déclenchées par un clignotement ou un flashage compris dans un intervalle de 4 à 59 flashs par seconde (Hertz) avec un pic de sensibilité à 20 flashs par seconde, ainsi que des changements rapides du sombre au lumineux (comme l'effet stroboscopique).

Figure 19. Recommandation 3.9 du WAI pour le clignotement d'écran

Plusieurs difficultés se posent au concepteur qui souhaite utiliser ces recommandations. D'abord, le jargon technique employé pour les décrire peut être un obstacle : le concepteur de la recommandation doit trouver le juste équilibre entre le jargon ergonomique opaque pour le non expert et le langage de vulgarisation qui ne décrit plus assez précisément les théories avancées. De ce problème de communication découlent des difficultés à interpréter une recommandation lorsqu'une ambiguïté apparaît dans sa définition : cette ambiguïté serait probablement levée en un clin d'œil pour un ergonome, mais ce sera plus délicat pour le novice qui a toutes les chances de mal appliquer la recommandation, faute de la comprendre correctement. Le grand nombre de recommandations disponibles est un autre obstacle à leur bonne utilisation. Le concepteur a besoin de guidage pour trouver, dans l'ensemble des recommandations proposées, celle qui correspond à son contexte de conception. Des travaux ont ainsi été menés pour une organisation et une diffusion pertinente de recommandations aux concepteurs (Scapin et coll. 2000). D'autres ont classifié les types de recommandations disponibles dans la littérature, pour guider le concepteur vers le choix d'un ou autre de ces types, en fonction du contexte de conception (Mariage et coll. 2005).

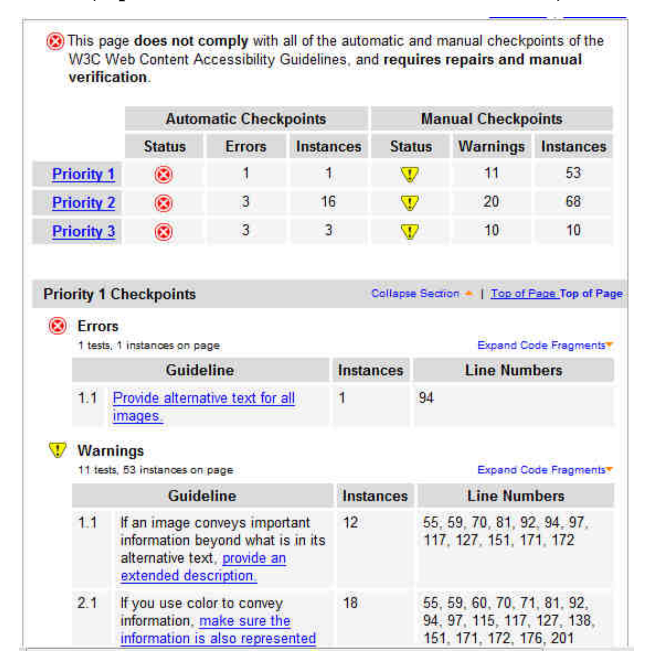
2.5.2. Outils d'évaluation ergonomique

Certaines règles ergonomiques peuvent être vérifiées automatiquement par inspection de l'application Web, et des outils de support à cette vérification existent pour le Web (Brajnik 2000). Parmi eux, WebTango (Ivory et Hearst 2002) compare automatiquement un site Web à des sites de même profil ayant été jugés d'un bon niveau d'utilisabilité par des experts humains. C'est un moyen de conserver les avantages de l'automatisation (coût, rapidité d'exécution, objectivité du rapport d'analyse), tout en injectant une dose d'expertise humaine à l'analyse.

L'outil WebXACT (Watchfire 2004) analyse quant à lui une page Web et renvoie un rapport sur la qualité d'accessibilité de cette page, pondéré par des niveaux de priorité, car toutes les recommandations n'ont pas la même importance. Un exemple de rapport d'analyse, réalisé sur la page d'accueil du site français de centralisation des changements d'adresse[1] (capture d'écran en Figure 20) est montré en Figure 21. Des recommandations figurent sur le rapport, ce qui fournit un support direct au concepteur pour améliorer la qualité de son application.



Figure 20. Page d'accueil du site français de changement d'adresse (capture d'écran réalisée le 24 décembre 2007)



This page **does not comply** with all of the automatic and manual checkpoints of the W3C Web Content Accessibility Guidelines, and **requires repairs and manual verification**.

	Automatic Checkpoints			Manual Checkpoints		
	Status	Errors	Instances	Status	Warnings	Instances
Priority 1	⊗	1	1	!	11	53
Priority 2	⊗	3	16	!	20	68
Priority 3	⊗	3	3	!	10	10

Priority 1 Checkpoints — Collapse Section | Top of Page Top of Page

Errors
1 tests, 1 instances on page — Expand Code Fragments

	Guideline	Instances	Line Numbers
1.1	Provide alternative text for all images.	1	94

Warnings
11 tests, 53 instances on page — Expand Code Fragments

	Guideline	Instances	Line Numbers
1.1	If an image conveys important information beyond what is in its alternative text, provide an extended description.	12	55, 59, 70, 81, 92, 94, 97, 117, 127, 151, 171, 172
2.1	If you use color to convey information, make sure the information is also represented	18	55, 59, 60, 70, 71, 81, 92, 94, 97, 115, 117, 127, 138, 151, 171, 172, 176, 201

Figure 21. Extrait du rapport WebXACT de la page d'accueil du site français de signalement de changement d'adresse (le 24 décembre 2007)

[1] https://www.changement-adresse.gouv.fr/ (dernier accès : 25 sept. 08)

2.5.3. Patrons d interface

Un patron de conception[1] est une solution à un problème de conception récurrent, qui a été prouvée scientifiquement ou empiriquement : sa validité a quoiqu'il en soit été éprouvée par la pratique, et la communauté reconnaît le bien-fondé de son raisonnement. Cette solution se veut concrète et directement applicable à un problème rencontré, moyennant une adaptation au contexte concerné. Cette méthode de support à la conception a été initiée dans le domaine de l'architecture, au travers de patrons de conception de bâtiments, de rues ou encore de jardins publics (Alexander et coll. 1977). En informatique, le problème traité par un patron de conception peut porter sur l'architecture logicielle (Gamma et coll. 1994), l'organisation de l'équipe de conception, ou encore l'interface utilisateur. C'est à ce dernier type de patrons de conception que nous nous intéressons ici : un patron d'interface répond à un problème de conception d'interface sur la base de la connaissance de la communauté en IHM, et, en fournissant une solution directement applicable, guide le concepteur vers de bonnes pratiques de conception.

Un obstacle à l'utilisation optimale de patrons est l'organisation des patrons au sein d'un catalogue qui rend la recherche difficile et, surtout, la multitude de sources, souvent redondantes, qui se présentent au concepteur (Dearden et Finlay 2006). Toutefois, la littérature ne propose pas, à notre connaissance, d'évaluation de l'utilisabilité des catalogues de patrons d'interfaces en situation réelle d'usage. Des évaluations de l'utilisabilité de patrons comme méthode de conception auprès d'étudiants en informatique ont permis de dégager des axes de réflexion à ce propos. Les anti-patrons (Brown et coll. 1998), erreurs récurrentes en conception formalisés sous forme de patrons, y sont par exemple déconseillés (Kotzé et coll. 2008). Les conseils de type positif s'avèrent plus faciles à appliquer mais aussi plus formateurs.

Web usability pattern set*[2] *(Perzel et Kane 1999)

Kimberly Perzel part du constat que beaucoup d'applications Web sont conçues par des personnes n'ayant aucune expertise en conception d'interface. D'autres applications Web sont développées par des équipes professionnelles, mais fortement pluridisciplinaires de part la nature même du Web (bases de données, graphisme, gestion de contenu éditorial). Au sein de ces équipes, des incompréhensions et des ambiguïtés apparaissent, faute d'une formation et d'un vocabulaire communs. Les auteurs proposent ici que l'interface soit la plateforme de communication de telles équipes, et que, tout en prenant soin de concevoir une application utilisable, les patrons d'utilisabilité soient un moyen de communication entre les intervenants. Les patrons issus de ces travaux concernent l'interface de l'application Web ou bien l'environnement de cette application Web avec, par exemple, des patrons concernant le navigateur de l'utilisateur (p.ex. marquage d'une page en favori, lisibilité de l'adresse URL). Comme suggéré par les observations de Michael Mahemoff (Mahemoff et Johnston 1998), les patrons concernant l'**Interface Utilisateur** sont distingués de ceux concernant le **Système**. Certains patrons concernent toutefois ces deux éléments. Chaque patron est décrit grâce aux attributs suivants : *Problème* (sous forme de question formulée du point de vue du concepteur), *Contexte*, *Classification* (Interface Utilisateur ou Système), *Avantages* (liste de raisons d'appliquer le patron), *Solution*, *Justification rationnelle*, *Contexte en sortie*, *Exemple* (liens vers des applications Web qui suivent les recommandations du patron), *Patrons associés* (apparaissant dans ce même catalogue).

[1] En Anglais : *design pattern*
[2] Ensemble de patrons d'utilisabilité pour le web

A pattern approach to interaction design[1] *(Borchers 2000)*

Jan Borchers utilise le principe du patron pour capitaliser de la connaissance concernant la conception. Partant de ce constat, la méthodologie qu'il met en place est pédagogique, avec le but de rendre le concepteur d'interface expert en IHM, plutôt que de diffuser des recommandations, ou de standardiser les interfaces. Ce travail rejoint ainsi l'un des critères d'évaluation de la qualité de patrons de Sally Fincher (Fincher et Windsor 2000) selon lequel les patrons doivent proposer des solutions, mais aussi suggérer de nouvelles solutions, venant résoudre d'autres problèmes que ceux qui sont déjà inventoriés. Le cas d'étude de ce travail est la conception de bornes interactives, utilisées dans un lieu public tel qu'un musée. Plus particulièrement, *WorldBeat* est une borne interactive munie de bâtons infrarouges et placée dans un musée d'art contemporain (Borchers 1997). Les visiteurs du musée peuvent appréhender les concepts de la musique au travers de jeux sur cette borne tels que la reconnaissance d'instruments sur une bande son ou l'utilisation d'instruments virtuels. La structure des patrons et le graphe des références entre patrons sont formalisés. Les patrons sont ainsi décrits par la série d'attributs suivants : *Nom*, *Degré d'universalité* (mesure à quel point ce patron peut être appliqué à d'autres contextes de conception), *Illustration* (photographie du contexte de déploiement du patron), *Problème*, *Avantages* (optimisation de la conception), *Exemples*, *Solution* (généralisation des exemples proposant une solution au problème de conception), *Diagramme* (résumé informel de la solution préconisée). Le graphe des patrons proposés pour la conception de *WorldBeat* est présenté sur la Figure 22.

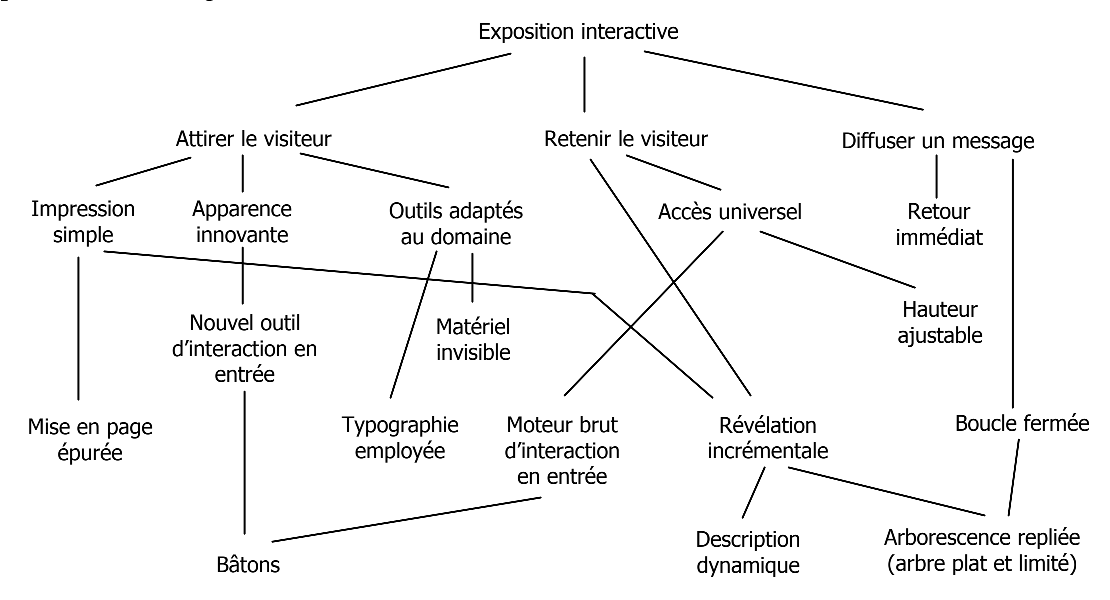


Figure 22. Graphe de patrons d'interaction pour les expositions interactives (issu de (Borchers 2000))

Experiences[2] *(Coram et Jim Lee 2002)*

Le catalogue de patrons *Experiences* (Coram et Jim Lee 2002) a été introduit pour résoudre des problèmes de conception d'interface de haut niveau. Les patrons sont reliés entre eux par un réseau de relations. A un mode d'interaction choisi correspond en effet un ensemble de patrons de plus petite taille, précisant l'implémentation de ce choix. Les interactions sont groupées sous un patron générique nommé **Style d'interaction** qui est détaillé en quatre patrons d'interaction correspondant à quatre choix de conception d'interface différents, quatre manières de proposer une interaction à l'utilisateur : **Formulaire d'entrée**, **Menu de sélection**, **Texte de conversation** et **Interface explorable** (cette dernière catégorie est décrite dans la vue présentée en Figure 23). Ce raffinement du grain de l'interaction par paliers prend modèle sur la structure

[1] Une approche basée sur des patrons pour la conception d'interfaces
[2] Expertises

proposée par Christopher Alexander (Alexander et coll. 1977) dans ses patrons de conception d'architecture. Les patrons eux-mêmes sont plutôt des recommandations, sans structure particulière ni liste d'attributs uniformisée à travers les patrons. Généralement, les recommandations fournies concernent l'expertise de l'utilisateur, les choix de conception à disposition du concepteur en fonction du contexte de l'application.

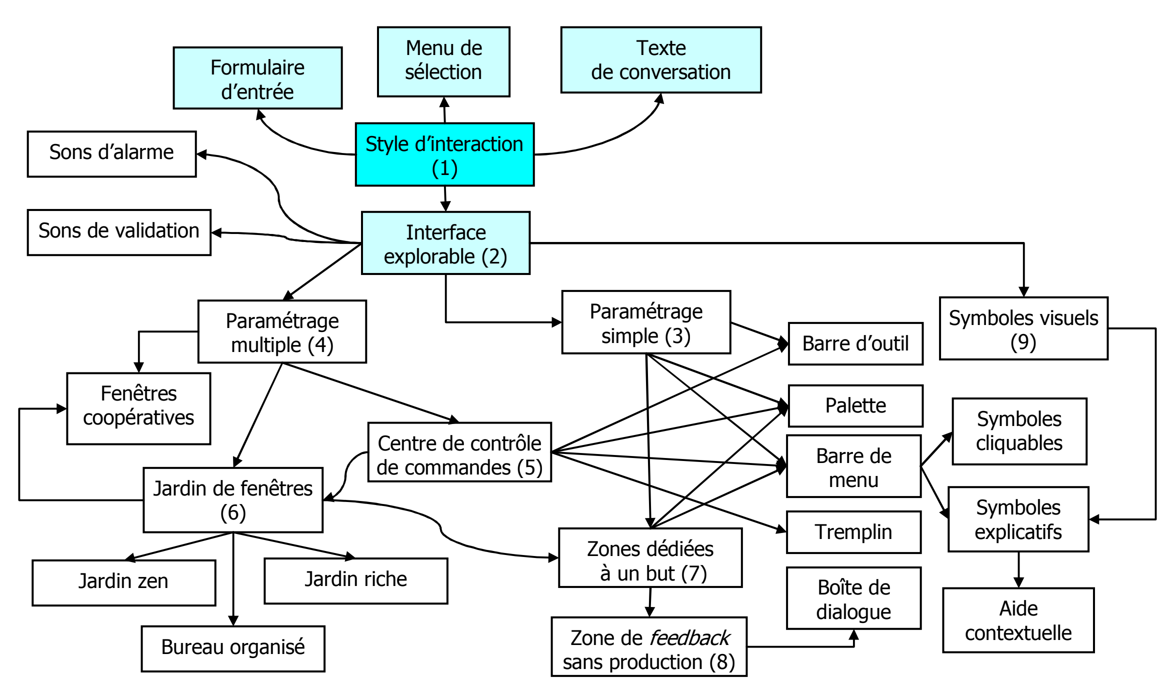


Figure 23. Vue partielle du réseau des patrons Experiences (issu de (Coram et Lee 2002))

Catalogue de patrons d'interfaces Web de (Montero et coll. 2002)

Ce catalogue propose des patrons dédiés à la conception d'applications web utilisables. Trois niveaux de spécification apparaissent, comme préconisé par Christopher Alexander (Alexander et coll. 1977), c'est-à-dire suivant le niveau de granularité de la question de conception d'interface considérée : **Niveau site web**, **Niveau page** et **Niveau ornementation**. De plus, comme décrit dans la Figure 24 ci-dessous, un réseau est tissé entre les patrons de même niveau, mais aussi entre patrons de niveaux différents. Ce tissage suit le sens de la réflexion lors de la conception en proposant d'associer à une question d'autres questions de granularité plus faible. La description des patrons eux-mêmes suit la structure suivante : *Motivation* (recommandation issue l'expertise IHM), *Problème* (formulation centrée utilisateur et/ou centrée application), *Avantages*, *Solution*, *Conséquences* (bénéfices de l'application du patron), *Exemples et Détails d'implémentation.*

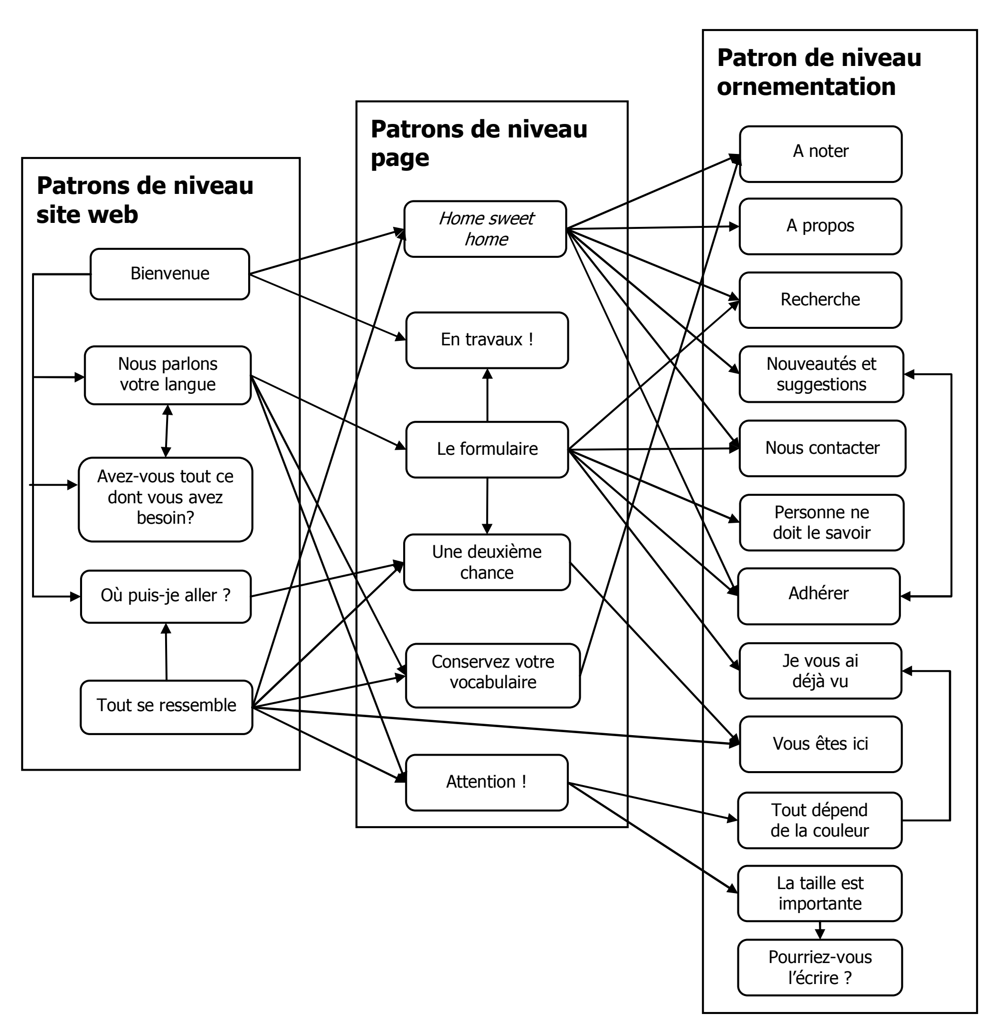


Figure 24. Catalogue de patrons proposé par Francisco Montero (issu de (Montero et coll. 2002))

The design of sites[1] *(Van Duyne, Landay, et Hong 2002)*

Le catalogue de Douglas K. Van Duyne est à l'usage des concepteurs d'interfaces de sites web, et il est organisé en plusieurs groupes correspondant à des besoins utilisateur formulés avec le point de vue du concepteur : **Natures de sites**, **Concevoir un cadre de navigation**, **Créer une page d'accueil efficace**, **Écrire et gérer du contenu**, **Instaurer de la confiance et de la crédibilité**, **e-Commerce basique**, **e-Commerce avancé**, **Aider les clients à réaliser leurs tâches**, **Concevoir des mises en page efficaces**, **Autoriser une recherche efficace et pertinente**, **Proposer une navigation facile** et **Accélérer le site**. Ces différentes catégories suivent également le cours du processus de conception, traitant d'abord de questions générales, puis de questions de plus en plus précises (d'abord le site, puis la navigation, puis la page, puis le contenu). Suivent ensuite des questions plus globales, nécessitant que la conception du site ait déjà atteint une certaine maturité, en termes de volume de contenu mais aussi de cadre de navigation (p.ex. instaurer la confiance, accélérer le site). Le contenu des patrons suit la structure suivante : *Captures d'écran d'exemples réels illustratifs*, *Contexte* (quels autres patrons peuvent appeler l'utilisation de ce patron), *Problème* (problème de conception d'interface, détaillé en problèmes plus précis relatifs à d'autres patrons éventuellement), *Solution* (recommandations concrètes pour la mise en place du patron, récapitulées dans un schéma informel), *Autres patrons à considérer*.

[1] La conception de sites

User Interface Design Patterns[1] *(Laakso 2003)*

Ces patrons de conception d'interface concernent plusieurs types d'applications, et pas seulement les applications web. Les recommandations portent sur la visualisation et sur les moyens d'organiser les informations. Quelques interactions fonctionnelles apparaissent également. C'est selon ces interactions de l'utilisateur, qu'elles soient à but informationnel ou à but fonctionnel, que les patrons de conception d'interface de Sari Laakso sont organisés : **Recherche**, **Vues de données**, **Stockages**, **Sélection et manipulation d'objets**, **Temps**, **Hiérarchies et ensembles**, **Enregistrer et annuler**. Le contenu des patrons est un ensemble de recommandations écrites en style télégraphique, avec un vocabulaire limité pour éviter les ambiguïtés. Les exemples qui illustrent ces recommandations proviennent d'articles scientifiques ou bien ce sont des exemples fictifs créés pour illustrer la mise en application du patron. Des références vers des publications scientifiques sont fournies en fin de patron, ainsi que des références vers d'autres patrons du catalogue.

Designing Interfaces[2] *(Tidwell 2005)*

Jenifer Tidwell a réalisé un catalogue de patrons de conception d'interfaces utilisateurs pour applications web, mais aussi pour d'autres types d'applications, en particulier les applications mobiles et les logiciels d'édition. Les catégories du catalogue renvoient à des tâches attribuées au concepteur de l'interface : **Organiser le contenu**, **Circuler** (navigation au sein de l'application), **Organiser la page**, **Faire des choses** (actions et commandes venant de l'utilisateur), **Montrer des données complexes** (arbres, tables, graphiques), **Récupérer de l'information saisie par l'utilisateur**, **Constructeurs et éditeurs** et **Soigner la présentation**. Ces catégories suivent un processus naturel de conception d'interface, en traitant dans un premier temps les questions liées au contenu, dans un deuxième temps les questions liées à l'interaction fonctionnelle avec l'utilisateur, et enfin les questions de graphisme. Les patrons sont décrits suivant la structure suivante : *Quoi* (résumé de la mise en place du patron), *Cas d'utilisation*, *Pourquoi* (argumentation rationnelle de l'utilisation du patron, basée sur des recommandations ergonomiques dans la plupart des cas), *Comment* (détails sur l'implémentation du patron, appuyés par un schéma informel explicatif), *Exemples* (captures d'écrans d'exemples réels, commentés et annotés).

YUI: The Yahoo! Design Pattern Library[3] *(Yahoo! 2007)*

Cette bibliothèque de patrons est maintenue par un groupe de concepteurs d'interface de Yahoo! Le processus de réflexion qui a mené la classification de ces patrons est disponible en ligne (Malone, Leacock, et Wheeler 2005). A l'heure actuelle (*Septembre 2007*), la classification est faite selon les buts de l'utilisateur de l'utilisateur de l'application : **Rechercher**, **Naviguer**, **Lire**, **Sélectionner**, **Interagir**, **Donner un retour d'expérience**, **Personnaliser**. Les patrons de Yahoo! sont des patrons de nature traditionnelle, c'est-à-dire qu'ils proviennent d'observations d'interfaces dans des cas réels, et de standardisation d'interfaces efficaces et utilisables. Ces patrons sont proposés, édités, publiés puis évalués par une équipe d'experts en utilisabilité, avant d'être soumis aux commentaires des utilisateurs des patrons eux-mêmes : plusieurs itérations permettent d'atteindre la « Façon Yahoo! », dernière étape de l'amélioration de patrons. Chaque patron contient les attributs suivants : *Description brève*, *Cas d'utilisation*, *Détails d'implémentation*, *Patrons associés*, *Argumentation rationnelle basée sur des recommandations ergonomiques* et *Conseils pour appliquer ce patron tout en respectant les normes d'accessibilité de Yahoo!*[4].

---

[1] Patrons de conception d'interface utilisateur
[2] La conception d'interfaces
[3] Bibliothèque de Patrons de Conception de Yahoo!
[4] http://help.yahoo.com/help/uk/accessibility/ (dernier accès : 25 sept. 08)

Interaction Design Pattern Library[1] *(Martijn Van Welie 2007)*

Martijn Van Welie (Martijn Van Welie 2007) propose un catalogue de patrons pour la conception d'interfaces qui couvre plusieurs types d'applications interactives : applications mobiles, applications de bureau (GUI) et applications web. Ces patrons sont groupés selon plusieurs catégories correspondant aux motivations possibles de la recherche et de l'application d'un patron. L'auteur considère que le concepteur d'une interface se pose des questions de conception en termes de besoins. Des questions plus générales à propos du contexte de l'application forment la troisième et dernière catégorie du catalogue.

- Besoins de l'utilisateur regroupe les patrons visant la satisfaction par l'interface d'un besoin de l'utilisateur. Parmi eux figurent : naviguer sans but précis, interagir avec des éléments d'interface basiques, rechercher, interagir avec des données, personnaliser, faire des achats, prendre une décision, saisir des informations, divers.
- Besoins de l'application regroupe les patrons visant la satisfaction d'un besoin de l'application en termes de communication avec l'utilisateur : capter l'attention, donner un retour sur les actions commises (feedback), simplifier l'interaction.
- Contexte de la conception regroupe les patrons visant la satisfaction de besoins plus diffus, ayant trait à la nature même de l'application, ou bien à des besoins auxiliaires aux besoins fonctionnels que satisfait l'application (par exemple, tout en faisant des achats, je consulte le choix d'autres clients et je prends part à une communauté virtuelle) : type de site, expérience utilisateur, type de page.

Les patrons comprennent les attributs suivants : *Problème* (formulation centrée utilisateur), *Solution*, *Cas d'utilisation*, *Support à l'implémentation*, *Justification rationnelle*, *Exemples*. Le catalogue est à disposition de ses utilisateurs sur un site web ; les utilisateurs peuvent déposer un commentaire à propos d'un patron, ou répondre à des commentaires déjà déposés.

2.5.4. Gestion de la connaissance ergonomique pour l'e-Gouvernement

Dans le cas de l'e-Gouvernement, les **recommandations ergonomiques** sont un moyen pour les institutions gouvernementales de définir précisément leurs exigences en matière d'utilisabilité et de fournir en même temps des moyens d'évaluer ces mêmes exigences sur des applications existantes. De même les règles ergonomiques constituent un format adéquat pour l'expression des standards de bonne conduite de conception, édités par les organismes de standardisation internationaux. Toutefois, les recommandations légales, notamment en ce qui concerne l'accessibilité, sont souvent formulées en des termes trop peu techniques et concrets (Agostini et Naggi 2007).

Malgré la règlementation de nombreux pays obligeant le respect de l'accessibilité pour les applications d'e-Gouvernement, et en dépit d'un effort de la part de la communauté Web pour produire des standards d'accessibilité (p.ex. W3C/WAI WCAG 1.0 (World Wide Web Consortium (W3C) 2007)), le niveau d'accessibilité en e-Gouvernement est encore imparfait (CapGemini 2006). Le problème vient d'une part d'une définition ambiguë de l'accessibilité dans les règlementations (Agostini et Naggi 2007), mais aussi d'un manque de support à l'interprétation des standards d'accessibilité existants. Pour encourager les concepteurs à se soucier d'accessibilité, des travaux s'attachent à fournir des **outils de vérification** intervenant de plus en plus tôt dans le processus de conception. Les travaux portant sur l'intégration automatique de règles ergonomiques en e-Gouvernement concernent ainsi majoritairement l'accessibilité. Soucieux de respecter les standards d'accessibilité, e-Citiz (Genitech 2007) est un éditeur d'applications d'e-Procuration qui fournit une vérification des règles d'accessibilité au cours de la conception. La spécification de l'application est basée sur des modèles (données, processus, navigation) et la vérification porte sur ces artefacts intermédiaires comme le recommandent (Xiong, Farenc, et Winckler 2007).

[1] Bibliothèque de Patrons de Conception de l'Interaction

Au vu de notre état de l'art de la littérature, il n'a pas été publié à l'heure actuelle de catalogue de **patrons** d'interface pour l'e-Procuration. Cette absence est sans doute due au fait que la maturité requise pour l'établissement de patrons a été atteinte très récemment dans le domaine : dégager de bonnes conduites de conception nécessite en effet un recul important sur les alternatives de conception et leurs conséquences comparées. En revanche, de nombreux patrons ont été définis pour l'e-Commerce (p.ex. (Martjin Van Welie 2007)), domaine proche de l'e-Procuration sur certains aspects (Wimmer 2001). Cependant, la nature différente de ces deux domaines, quoiqu'ils aient l'aspect procédural en commun, fait émerger bien vite de capitales différences du point de vue de l'interface. Un patron d'interface pour l'e-Commerce inclura des éléments d'incitation à l'achat, tandis qu'un patron de page d'e-Procuration, loin de songer à susciter de nouveaux besoins chez l'utilisateur, proposera une réponse formelle et claire aux seuls besoins qui l'ont poussé à visiter l'application. De plus, la définition de patrons doit venir du contexte d'utilisation, de l'expérience de conception du domaine, d'observations acérées sur les conditions de conception : cela rend bancale l'utilisation pour l'e-Procuration de patrons prévus pour d'autres domaines.

2.6. SYNTHÈSE ET DISCUSSION

Les méthodes d'aide à la conception étudiées dans les sections précédentes dans ce chapitre servent de guide et de support à l'inclusion des besoins et contraintes des utilisateurs dans la conception de l'interface. Elles offrent des modes opératoires à suivre pour recueillir ou formaliser les besoins utilisateurs ou évaluer la façon dont ces besoins sont satisfaits grâce à l'application. Ces méthodes produisent des artefacts de différentes sortes :

- modèles de tâche, de procédure, de domaine, d'organisation, de navigation ;
- rapports d'étude de l'utilisateur au travers d'observations, de séances de conception participative, de définition de *Personas* ou de groupes de discussions ;
- prototypes de l'interface ;
- rapports d'évaluation ergonomique de l'interface réalisés grâce à l'observation de l'utilisateur, l'inspection de règles ou automatiquement grâce à un outil.

La conception d'une application Web comporte une grande part de discussions contradictoires pour l'émergence de solutions satisfaisantes. La réalisation des artefacts de spécification évoqués plus haut donne lieu à des discussions au sein des membres de l'équipe de conception qui y participent. Une fois l'artefact réalisé, l'équipe de conception en tient compte pour les étapes suivantes de la conception (p.ex. modification d'éléments non ergonomiques de l'interface, développement d'un prototype haute fidélité, adaptation du vocabulaire de l'application d'après les remarques émises en groupe de discussion). Ces deux étapes de réalisation de la méthode et d'adaptation de la conception aux conclusions tirées constituent deux occasions pour tout ou partie de l'équipe de conception d'échanger ses points de vue et de discuter des suites de la conception de l'application. Dans le cadre de l'e-Procuration, une méthode d'aide à la conception de l'interface doit suivre les principes listés ci-dessous et produire des artefacts qui suivent également ces principes.

- La méthode offre un support à la communication au sein de l'équipe :
  - L'artefact produit par la méthode est **lisible** par l'ensemble des membres de l'équipe de conception ;
  - L'artefact produit par la méthode décrit une solution ou un problème de façon **non ambiguë** ;
  - L'artefact produit par la méthode peut servir de support à **l'argumentation** : l'invoquer peut appuyer ou contrer un argument ;

– L’artefact produit par la méthode peut servir de support à la **recherche d’un consensus** : en le modifiant, plusieurs personnes en désaccord peuvent parvenir à une solution satisfaisante pour chacun ;

- La méthode offre un support concret à la conception d’interface ; en cas de désaccord, la méthode aide à trancher en faveur du choix le plus pertinent pour l’utilisateur :
  – La **conception graphique** est facilitée : choix des couleurs, des polices de caractères (fonte et taille des caractères), disposition des pages (p.ex. deux colonnes), apparence des composants interactifs (p.ex. boutons de formulaire)… ;
  – L’**édition du contenu** est facilitée : quantité et répartition optimales des informations, vocabulaire à utiliser pour un guidage optimal de l’utilisateur… ;
  – La **conception de la navigation** est facilitée : un support est fourni pour la définition de la structure hypertexte ;
- La méthode produit des artefacts qui peuvent être capitalisés pour de futurs projets
  – Les artefacts produits par la méthode constituent des **solutions pérennes** pour le domaine de l’e-Procuration ;
  – La méthode offre un **cadre de stockage** pour les solutions développées.

Tableau 3. Comparaison du potentiel des méthodes étudiées pour la conception d'interfaces en e-Procuration

		Modélisation des tâches utilisateur	**Modélisation de la navigation**	**Modélisation de la procédure**	**Modélisation du domaine**	**Analyse de l’utilisateur**					**Règles ergonomiques**	**Outils d’évaluation automatique**	**Patrons d’interface**
						Observ. des utilisateurs	Conception participative	Prototypage basse fidélité	*Personas*	Groupes de discussion			
COMMUNICATION	Lisibilité	✓	✓	✓	~	✓	~	✓	✓	✓	~	~	✓
	Non ambiguïté	✓	✓	✓	✓	×	~	✓	×	×	~	✓	~
	Support à l’argumentation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Support à la recherche de consensus	✓	✓	~	~	×	✓	✓	×	×	✓	✓	✓
CONCEPTION	Support à la conception graphique	✓	×	~	×	~	~	~	×	~	✓	~	✓
	Support à l’édition du contenu	×	×	~	✓	~	~	~	~	~	~	~	~
	Support à la conception de la navigation	~	✓	~	×	×	~	~	×	~	✓	~	✓
CAPITALISATION	Pérennité des solutions développées	✓	✓	~	~	×	×	~	~	×	×	×	✓
	Extensibilité et cadre de stockage	×	×	×	×	×	×	×	~	×	✓	×	✓

Ces critères sont repris dans le Tableau 3 ci-dessus pour évaluer le potentiel des méthodes listées dans cette section pour la conception d’une application d’e-Procuration. Deux méthodes

d'une même catégorie couvrent différemment un ensemble de critères donnés, selon le profil scientifique des auteurs de la méthode et leurs objectifs. Nous avions pu constater cette disparité entre méthodes dans le Tableau 1. Adéquation aux besoins de l'e-Procuration des méthodes de spécification de systèmes interactifs étudiées. Les critères sont ici évalués sur les catégories de méthodes de façon générale, selon la couverture moyenne des méthodes de la catégorie, mais aussi selon la raison d'être initiale de la catégorie de méthode considérée. Cette évaluation globale nous a permis de dégager des pistes sur le bien-fondé de l'emploi de ces méthodes pour la conception en e-Gouvernement. Ces pistes sont présentées ci-après, selon les trois besoins dégagés pour la conception en e-Gouvernement 1) Favoriser la communication, 2) Soutenir la conception et 3) Capitaliser les connaissances.

2.6.1. Favoriser la communication

Les **modèles** étudiés dans cette section peuvent être lisibles par tout membre de l'équipe de conception, si le formalisme est choisi selon un critère de lisibilité et non ambiguïté, et moyennant une formation à la lecture et à l'édition simple *a minima* de ces modèles. En tant que description formelle d'une partie de l'application, ils peuvent servir de support à la discussion, et être modifiés de façon coopérative pour atteindre une solution consensuelle.

Le **prototypage** basse fidélité intervient souvent dans le cadre de conceptions participatives et constitue un moyen efficace de recueillir les remarques des utilisateurs aussi bien que de l'ensemble des membres de l'équipe de conception, quelle que soit leur expertise.

En revanche, les rapports d'**observations** ou de **groupes de discussion**, en plus d'être soumis à la subjectivité de leurs rédacteurs, offrent un volume d'information qui peut désorienter les personnes inexpertes en facteurs humains. L'exploitation de telles études exige un recul sur la situation observée, et, hormis sur des points statistiques précis (p.ex. taux de satisfaction moyen de l'utilisation de la section *S* de l'application), les seuls intervenants aptes à utiliser les résultats sont des experts en facteurs humains.

Les moyens de matérialiser l'utilisateur sous forme de ***Persona*** servent à l'argumentation, mais sont trop abstraits et ambigus pour permettre la recherche d'une solution consensuelle qui satisfasse tous les intervenants. L'utilisation d'une personne fictive, même si sa description est détaillée et cohérente, offre toujours des zones floues. Ces zones mal définies de la *Persona* ne serviront pas de support à des discussions sur des points de conceptions très précis, faute de cohésion entre les points de vue que les intervenants ont sur la *Persona*.

2.6.2. Soutenir la conception

Tout ce qui permet d'évaluer une solution (**conception participative, prototypage basse fidélité, groupes de discussion, outils et méthodes d'évaluation**), constitue un support consultatif à la conception, et des supports efficaces de discussion : ils sont précieux mais insuffisants pour guider l'équipe de conception de manière concrète. Ils ne fournissent pas de règles de bonne conduite à suivre pour s'assurer du bien-fondé d'une décision de conception d'interface.

Les **modèles** s'attachant à la description d'une partie de l'application doivent être combinés pour constituer un véritable support à la conception (cf. §2.3.3). Le développement d'une architecture de conception basée sur plusieurs modèles serait un support pertinent à la conception d'applications d'e-Procuration (Pontico, Farenc, et Winckler 2006). Cependant, sans cadre méthodologique bien structuré, un tel mode de conception requiert la présence d'experts des différents domaines de modélisation dans l'équipe de conception, ce qui est une hypothèse utopique et qui exclut la plupart des membres de l'équipe de conception du travail de réflexion.

De manière générale, excepté les modèles qui aident à la conception d'une partie de l'application, les méthodes étudiées s'avèrent faibles en termes de support concret à la conception. Les **patrons d'interface** peuvent quant à eux être d'une utilité remarquable, si leur contenu est conforme à ce qui est nécessaire dans le domaine d'application, ce qui n'est pour le moment pas le cas en termes de support à l'édition de contenu. Ils pourraient proposer un support

efficace en fournissant des éléments d'information et de description standard. Le manque d'efficacité des patrons d'interface vient donc de leur manque d'adaptation au contexte d'e-Procuration qui nous intéresse ici.

2.6.3. Capitaliser la connaissance

Les **observations** trop circonstanciées ne sont pas réutilisables, car le cadre de l'e-Gouvernement, et notamment son acceptation chez les citoyens, fluctuent très rapidement. L'environnement des agents administratifs évolue rapidement également. Notamment l'introduction d'une nouvelle application modifie significativement l'environnement d'utilisation : l'étude réalisée pour l'introduction d'une première application sera donc périmée au moment d'en introduire une seconde. Cependant, certaines ***Personas*** représentant des citoyens peuvent être réutilisées d'une application sur l'autre. L'exemple de *Persona* « Matthieu » proposé dans la Figure 11 peut ainsi être invoquée dans la conception de toute application d'e-Procuration destinée à un citoyen français en activité : déclaration de revenus, changement d'adresse, gestion de compte d'assuré social, etc. Il peut d'ailleurs être judicieux de croiser entre projets ou équipes de conception ce genre de *Persona* « transversale ». Une *Persona* citoyenne inventée pour une application présentera des caractéristiques inattendues pour l'équipe de conception d'une autre application, et l'exécution de scénarios sur la base de cette personne fictive sera une évaluation intéressante. Ce croisement de *Personas* aidera les équipes de conception à envisager le large champ des caractéristiques possibles pour un citoyen. Pour ce faire, toutefois, un cadre de définition précis et complet des *Personas* doit être agréé par tous les projets participants, pour plus de lisibilité, moins d'ambiguïté et une information suffisante pour plusieurs domaines d'application dans l'e-Gouvernement. La vie privée de la *Persona* devra être renseignée en détails (situation familiale, nombre d'enfants, ville de résidence, projets personnels), ainsi que sa vie professionnelle (formation, profession, statut, projets professionnels) ou encore sa vie citoyenne (implication dans la vie de la cité, adhésion à une association, habitudes de participation à la vie citoyenne).

Pour la capitalisation des connaissances, le meilleur moyen de stockage et de réutilisation concrète des savoir-faire est le **patron de conception d'interface**. Cette méthode permet de formaliser les connaissances emmagasinées, sous une forme directement applicable. Toutefois, le cadre de classement des patrons et leur composition doivent être étudiés pour le cas particulier d'une application d'e-Procuration. Quelles informations sont nécessaires pour qu'un patron d'interface d'e-Procuration soit applicable ? Quelle latitude faut-il laisser pour la personnalisation d'une solution standard ? Comment établir un standard d'interface qui garantisse l'utilisabilité sans être un frein à la créativité ? Quand et comment intégrer l'utilisation de patrons d'interface à un processus de conception d'e-Procuration ? Comment fournir les moyens d'explorer les patrons pour que le concepteur reconnaisse, parmi les solutions proposées, celle qui correspond au problème rencontré ? Comment fournir des moyens de comparer plusieurs solutions entre elles ? Comment autoriser l'ajout d'une solution n'apparaissant pas dans le catalogue de patrons d'interface ? Quelle structure mettre en place pour valider et publier cette nouvelle solution comme une nouvelle bonne pratique de conception d'interface d'e-Procuration ? Nous tentons de répondre à ces questions dans la section suivante qui présente notre contribution à une méthode de conception d'interfaces d'e-Gouvernement basée sur des patrons.

3. L'e-Gouvernement sur le terrain, une observation exploratoire

Résumé

Une observation naturaliste des activités de conception a été menée dans l'entreprise SmalS, dédiée au développement d'applications pour le gouvernement belge. Il s'est agi de confronter les méthodes théoriques de conception telles qu'étudiées dans le chapitre 2 aux conditions réelles de travail des concepteurs d'applications d'e-Gouvernement. Les méthodes de conception et de spécification en particulier, les modes de communication, les intervenants et leurs interactions, les projets passés et leur devenir et les interfaces des applications existantes ont fait l'objet d'observations naïves, pour faire émerger les forces et les faiblesses de ce cadre particulier de travail. Ces activités ayant révélé la nature très récursive de l'e-Procuration, l'hypothèse d'une méthode de conception basée sur des patrons d'interface pour l'e-Procuration a été étudiée en profondeur.

Nombreuses sont désormais les entreprises de technologies de l'information qui répondent à des appels d'offre publics concernant des projets d'e-Gouvernement, notamment en Allemagne, en Finlande et en France (CapGemini et TNO 2004). Elles apportent leur compétence en matière de gestion de workflows complexes, familière aux entreprises gestionnaires d'applications bancaires par exemple. Certaines de ces entreprises rédigent des rapports sur l'état d'avancement des problématiques de conception en e-Gouvernement (Accenture 2005) (CapGemini 2006) mais peu de retours sur leurs activités réelles sont disponibles. Des travaux académiques ont toutefois établi des protocoles d'observation des activités de conception en e-Gouvernement dans un cadre expérimental de conception en équipe (Halstead-Nussloch et coll. 2003).

Le faible volume d'informations disponible sur les activités réelles de conception en e-Gouvernement a nourri notre besoin d'une observation sur le terrain. L'observation exploratoire des activités de conception chez SmalS[1] s'est déroulée durant quatre semaines, en octobre 2006. L'objectif était de compléter notre étude théorique de la conception centrée utilisateur en e-Gouvernement par des données concrètes, issues de la conception en milieu industriel. Ce séjour a ainsi été l'occasion de collecter des études de cas réelles et d'observer les activités de conception (coordination des activités, méthodes et outils utilisés).

Notre séjour chez SmalS a coïncidé avec les débuts de la mise en place d'un outil de support à la conception centrée utilisateur : le StyleGuide. Le centre de compétences en utilisabilité de SmalS avait observé plusieurs caractéristiques dans les activités de conception qui laissaient à penser que les méthodes employées étaient perfectibles :

- Qualité inégale des interfaces produites en termes d'utilisabilité ;
- Utilisation marginale, par les concepteurs, des recommandations ergonomiques mises à leur disposition par l'intermédiaire des formations du centre de compétences en utilisabilité ;
- Habitude prise par les développeurs de réutiliser des fragments de code d'une application, à plus ou moins bon escient. On trouve ainsi des fragments d'interfaces similaires (p.ex. page ou formulaire) sur des applications indépendantes les unes des autres.

Sur la base de ces observations, le centre de compétences en utilisabilité décide en 2006 de mettre en place un support à la capitalisation de bonnes pratiques de conception ergonomique

[1] http://www.smals.be (dernier accès : 25 sept. 08)

en e-Gouvernement. Ces bonnes pratiques guideraient les analystes et les développeurs d'interface vers des solutions standardisées de qualité. C'est dans ce cadre, accueillis au sein du centre de compétences en utilisabilité, que nous avons mené notre observation naïve du domaine au travers de l'entreprise IT SmalS et de ses expériences. Les activités d'observation menées ont été de nature participante, et nous avons pris part aux études préliminaires à la mise en œuvre du StyleGuide. Les activités suivantes nous ont ainsi permis de dégager les observations qui sont décrites dans la suite de ce chapitre :

- Observation du processus de conception et des activités de conception par des entretiens informels avec des membres de l'entreprise, sur leurs activités quotidiennes, les retours d'expérience sur les projets passés (cf. §3.1) ;
- Étude de la littérature au sujet de problématiques liées à la conception basée sur des patrons, et notamment leur organisation (cf. §3.2) ;
- Analyse des fragments récurrents d'interface dans les applications existantes, applications du projet de la Sécurité Sociale en particulier, recueil de bons et de mauvais exemples pour alimenter la définition des bonnes pratiques (cf. §3.3) ;

3.1. Étude de l'activité de conception

SmalS est une entreprise de technologies de l'information, basée à Bruxelles en Belgique. Cette entreprise à but non lucratif est dédiée au développement d'applications pour le gouvernement belge. Dès sa création, en 1939, SmalS se concentre sur des activités de gestion de l'information pour le secteur public, dans le domaine des allocations familiales à l'époque. D'année en année, le secteur se déploie avec de nouveaux supports technologiques, et intègre les possibilités offertes par Internet, au début des années 1990. Les projets les plus significatifs menés dès lors dans le domaine de l'e-Gouvernement concernent les applications suivantes :

- Le portail de la sécurité sociale[1] qui offre des dizaines de transactions électroniques pour les citoyens, les entreprises et les professionnels du secteur social ;
- La plateforme *be health*[2] qui diffuse des informations et met à disposition des transactions en ligne pour les professionnels de santé ;
- Le site Web « Au travail[3] » qui fournit des informations sur les mesures gouvernementales belges pour l'emploi, consultables du point de vue du demandeur d'emploi ou du point de vue de l'employeur potentiel ;
- L'application Limosa[4] qui permet à un employeur de déclarer en ligne un travailler étranger auprès de la sécurité sociale belge.

Avec plus de 1470 collaborateurs dont 980 informaticiens, SmalS est une structure de grande taille. En plus de services administratifs et techniques directement opérationnels, plusieurs pôles d'expertise et d'innovation concernent notamment l'utilisabilité, la sécurité, la gestion de documents dématérialisés ou la recherche d'informations. Nombreux sont les employés qui proviennent du milieu académique et qui continuent, en parallèle de leur travail industriel chez SmalS, à officier en tant qu'enseignants et chercheurs dans des laboratoires universitaires.

Cette section s'attache à décrire le mode suivi pour l'observation des activités de conception de cette entreprise, puis les données recueillies : intervenants dans le processus de conception et cycle de vie d'une application d'e-Procuration développée chez SmalS.

[1] https://www.socialsecurity.be/ (dernier accès : 25 sept. 08)
[2] Littéralement « Soyez santé », https://www.behealth.be/ (dernier accès : 25 sept. 08)
[3] http://www.autravail.be/ (dernier accès : 25 sept. 08)
[4] http://www.limosa.be/ (dernier accès : 25 sept. 08)

3.1.1. Intervenants dans le processus de conception

L'observation exploratoire menée chez SmalS nous a permis de constater la coopération de plusieurs types d'intervenants particuliers au domaine de l'e-Gouvernement. Les clients des projets sont des institutions. Les utilisateurs finaux sont des administrés ou des agents administratifs. L'équipe de développement est composée d'experts en graphisme et implémentation. Nous avons regroupé dans la catégorie « médiateurs » les intervenants au processus de conception qui sont aptes à jouer le rôle d'intermédiaires entre ces différents corps de métier qui ne partagent pas le même jargon, les mêmes contraintes et besoins vis-à-vis de l'application.

Client du projet

Le client d'un projet d'e-Gouvernement est un ou plusieurs responsables d'institutions qui commanditent la dématérialisation d'une procédure administrative sur le Web ou l'amélioration d'une application existante. Dans le cadre d'institutions de grande taille (p.ex. la sécurité sociale, un ministère), l'institution est représentée par un membre du service communication. Dans une institution de plus petite taille (p.ex. une école, une ville de moins de 5000 habitants), le responsable de l'institution intervient généralement en personne. Le client est en mesure de communiquer à l'équipe de conception les valeurs à transmettre via le graphisme de l'application : une institution souhaitera profiter d'un projet d'e-Gouvernement pour dépoussiérer son image et transmettre une valeur de dynamisme, tandis que d'autres préfèreront mettre en avant la robustesse de leur mode de fonctionnement, pour rassurer leurs utilisateurs. Le client définit aussi, avec l'aide de ses interlocuteurs, les fonctionnalités à faire apparaître ainsi que les données alimentant le contenu éditorial.

Dans les projets menés par SmalS, le chef de projet et les analystes rendent visite au client en début de conception afin de collecter les besoins fonctionnels et les contraintes financières et temporelles. Il arrive que le client demande à rencontrer le consultant en utilisabilité pour mieux connaître les utilisateurs, leurs attentes et leurs opinions. En phase de prototypage de l'interface, le client est ponctuellement invité à des réunions de conception afin de donner son avis sur les orientations choisies. La prise en compte de cet avis est discutée en sa présence.

Utilisateurs

Les utilisateurs d'une application d'e-Gouvernement sont des citoyens, des responsables d'association, des responsables d'entreprise ou des secrétaires. Ils peuvent également être des agents administratifs, travailleurs sociaux ou des employés du secteur public chargés du suivi d'une procédure. Dans les deux cas, la participation de l'utilisateur au processus de conception est primordiale, pour que l'application soit utile et utilisable, et pour des raisons spécifiques à chacun de ces deux publics.

Pour intégrer les utilisateurs au processus de conception de SmalS, lorsque ces utilisateurs sont identifiables et accessibles, un expert en utilisabilité recueille leurs besoins en tout début de conception. Le cycle de développement est basé sur des prototypes d'interface, sur lesquels des itérations sont réalisées jusqu'à obtention d'un consensus (cf. §3.1.2). Des tests utilisateurs sont menés sur ces prototypes, sous la direction d'un expert en facteurs humains, lorsque, au début des itérations, plusieurs alternatives sont difficiles à départager, ou en fin d'itérations, lorsqu'un prototype satisfaisant est obtenu, pour le valider et repérer d'éventuels problèmes d'utilisabilité. Les rapports entre SmalS et les utilisateurs se prolongent après le déploiement de l'application, puisque c'est l'entreprise qui est en charge de l'assistance téléphonique et de la maintenance.

Équipe de développement

L'équipe technique est formée d'un ou plusieurs représentants des rôles indiqués ci-dessous. De façon générale, ils travaillent de façon très occasionnelle en contact direct avec les clients et les utilisateurs. Dans ces occasions, ils sont accompagnés de médiateurs.

Le graphiste prend en charge l'aspect visuel de l'application. L'assemblage des couleurs, les formes apparaissant sur l'interface, les logos contribuent à la mise en place d'une ambiance qui doit s'accorder avec les valeurs des institutions concernées par l'application. Son expertise lui permet de satisfaire une partie des besoins en utilisabilité, en particulier pour ce qui concerne la lisibilité et l'emploi d'icônes et de couleurs évocatrices.

Le développeur d'interfaces est chargé de l'implémentation des éléments de l'application avec lesquels les utilisateurs interagissent, en sortie (affichage d'informations) ou en entrée (saisie d'informations, activation d'un lien hypertexte). Il travaille de façon étroite avec le développeur *back-end* et avec le graphiste pour que chacun s'assure de la compatibilité de ses décisions avec celles des autres, et de leur faisabilité technique. Son travail de réflexion se base sur les patrons existants, il participe activement à leur personnalisation et donne son avis pour leur gestion et leur mise à jour, pour ce qui concerne leur faisabilité technique.

Le développeur *back-end* est chargé de l'implémentation de l'application *back-end*, c'est-à-dire le système dorsal de développement, derrière l'interface, invisible pour l'utilisateur ; y sont par exemple implémentés l'accès à une base de données, le calcul de variables, la génération des données dynamiques à afficher sur une page. Il ne participe pas aux réunions générales de conception, mais il y est représenté par le responsable des développeurs qui est capable d'évaluer la faisabilité des choix de conception réalisés.

Les experts en bases de données, sécurité, cryptographie, ou encore en réseaux participent également au développement. Ils ont un rôle de consultation et participent plus ponctuellement que les graphistes ou développeurs. Leur activité quotidienne est une activité de recherche pour la mise en place de systèmes innovants dans leur domaine.

Médiateurs

Les médiateurs assurent un lien entre les institutions à l'initiative de l'application et les utilisateurs d'une part, et l'équipe technique d'autre part. Ils ont un rôle de passerelle et connaissent les jargons, les intérêts, les besoins et les contraintes de plusieurs corps de métier impliqués dans la conception. Les médiateurs cherchent et suggèrent des consensus qui satisfont chaque partie (p.ex. le client et le développeur ; deux institutions partenaires) tout en garantissant que la décision n'ira pas à l'encontre de la qualité de l'application.

Le chef de projet est le responsable de la bonne conduite du projet, le garant de la tenue des délais et des contraintes financières mais aussi de la satisfaction du client et, par le biais du client, de la satisfaction des utilisateurs de l'application. Il donne un avis consultatif au fil de la conception, en particulier sur la conception des pages de l'application, et de leur contenu éditorial.

L'analyste est l'interlocuteur privilégié du client, c'est-à-dire de l'institution demandeuse de l'application, en tout début de conception. Fort de son aptitude à dialoguer avec le client avec un vocabulaire approprié, de sa connaissance des contraintes matérielles du projet et de sa conscience des solutions techniques envisageables, il recueille les besoins du client. Sur leur base, il réalise la spécification de l'application, en contrôle la validité avec le client, et le communique à l'équipe de projet pour démarrer la conception.

L'expert en utilisabilité est un spécialiste des facteurs humains qui possède également de solides connaissances en développement d'interfaces. Cette double compétence lui confère une expertise en matière d'ergonomie pour l'interface. Il est également le lien entre l'équipe de projet et les utilisateurs, étant en charge du recueil des besoins utilisateurs et de la conduite des tests utilisateurs. Il participe à la conception d'interfaces en faisant référence aux tests conduits, aux besoins collectés, et également à son expertise théorique en facteurs humains.

Le responsable éditorial est un membre de l'équipe technique mais il a une connaissance aiguisée des concepts du domaine administratif. Il est chargé de la relecture du contenu des pa-

ges, et en tant que chargé de ce contenu, il a une grande responsabilité dans la lisibilité des pages.

3.1.2. Cycle de développement et méthodes de conception employés

Le cycle de développement suivi dans les projets dirigés par SmalS est schématisé sur la Figure 25. Il s'agit d'un cycle de développement itératif, basé sur des prototypes basse puis haute fidélité pour la conception de l'interface. Lorsque le prototype atteint un stade satisfaisant pour les intervenants au processus de conception, y compris les utilisateurs et les clients, l'application elle-même est implémentée puis déployée. Les étapes successives de ce cycle de développement sont détaillées ci-après.

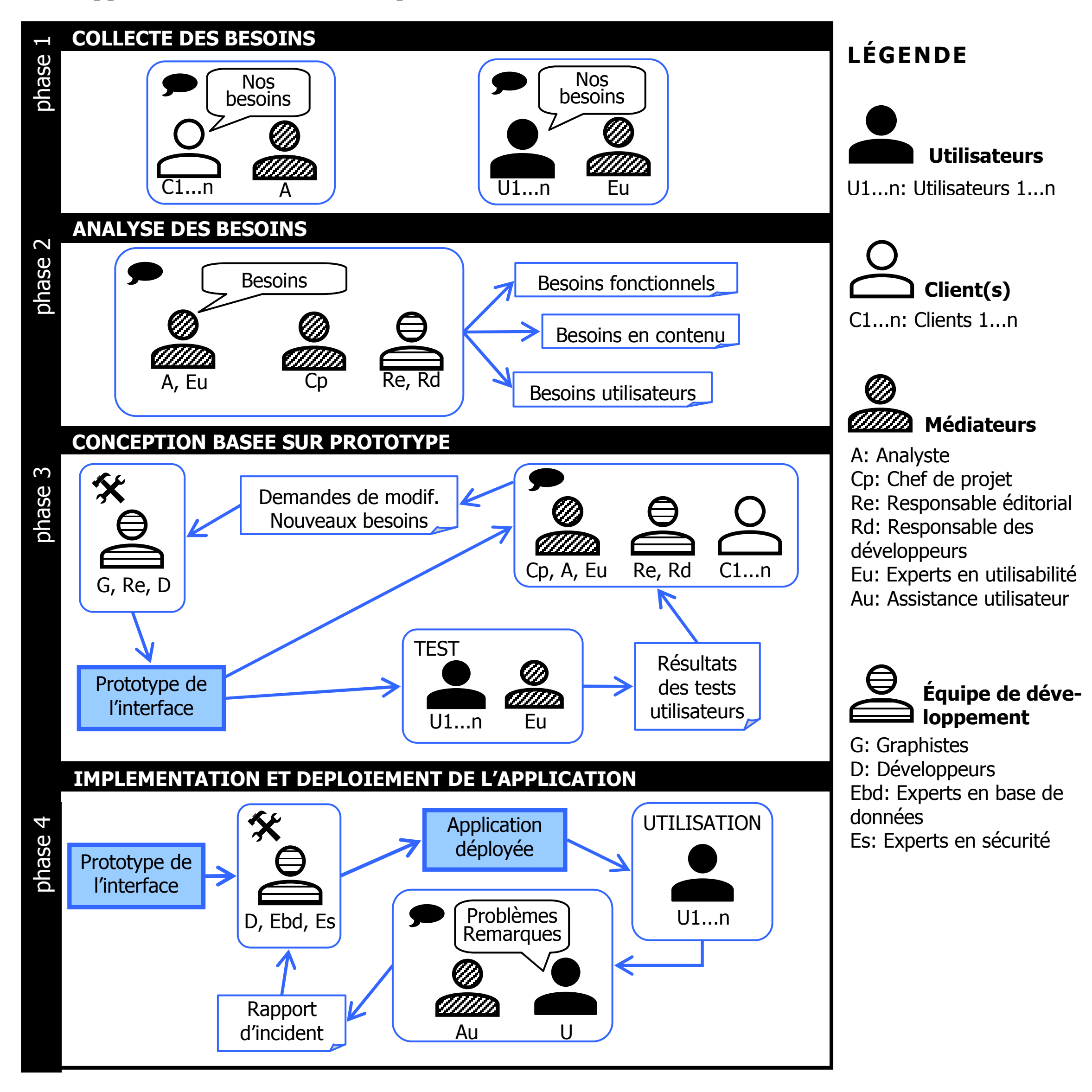


Figure 25. Exemple de cycle de développement d'une application d'e-Gouvernement : l'entreprise SmalS (Bruxelles, Belgique)

Collecte des besoins

Le client évoque ses besoins fonctionnels lors de discussions avec des analystes, avec comme point de départ la motivation du client pour ce projet, plus ou moins précise : dématérialisation d'une procédure spécifique par le biais d'une application d'e-Procuration, refonte d'une application existante, établissement de partenariats entre institutions pour des procédures croisant leurs services. Le but peut être la modernisation générale de l'image de l'institution,

l'accélération du traitement des procédures ou encore l'amélioration de l'utilisabilité de l'application. La synthèse de ces collectes se fait sans support particulier, si ce n'est d'éventuelles listes de pointage provenant de la culture théorique des analystes et de leur expérience dans le domaine. Lorsqu'ils ont une expérience de collecte des besoins en e-Gouvernement, les analystes acquièrent en effet des réflexes et connaissent les points délicats sur lesquels la réflexion avec le client doit être approfondie.

Les utilisateurs, dans les cas où ils sont identifiés et accessibles, rencontrent des experts en utilisabilité, pour que leurs besoins soient collectés. Dans le cas d'une application à destination des citoyens, public très large et sans représentant véritable, la collecte des besoins se fera par une étude des procédures existantes ou des questionnaires, sans rencontre directe des utilisateurs. En revanche, dans le cas où les utilisateurs finaux sont des employés de l'institution demandeuse, agents administratifs ou travailleurs sociaux, la collecte de leurs besoins se fait directement auprès d'eux, par entretiens et observations de leurs conditions d'utilisation de l'application.

Analyse des besoins

Dans cette phase, les besoins collectés sont rapportés par les analystes et les experts en utilisabilité à une partie de l'équipe de conception, représentative de chaque service : chef de projet, responsable éditorial et responsable des développeurs. Les besoins recueillis dans la phase précédente sont décrits sans support formel, oralement, avec le support d'extraits des études menées, entretiens ou observations. Le prototypage HTML est fréquemment utilisé pour l'expression des besoins par les analystes, et ce, de façon importune, de leur propre aveu. Sur la base des besoins collectés auprès du client, ils proposeront ainsi un prototype d'interface réalisé avec le langage HTML, plutôt que de choisir le support du papier ou d'outils d'édition de diagrammes par exemple. Plusieurs analystes ont approuvé la déclaration de l'un d'eux, au cours d'une réunion :

> « Il est tentant pour moi, d'exprimer les besoins directement par le biais de pages HTML, et j'utilise cet outil quasi systématiquement. Le codage est facile, même si je n'ai pas de formation en développement informatique, et il me fournit rapidement un support expressif et lisible pour mes interlocuteurs. Toutefois, je me rends compte que par ce biais, je ne rapporte pas seulement les besoins, je les interprète et je sous-entends des décisions anticipées en termes d'interface. »
>
> -- Un analyste, Réunion de conception du StyleGuide, octobre 2006.

Les besoins sont discutés, ajustés et détaillés en fonction des contraintes techniques, économiques et temporelles de chaque membre de l'équipe et des personnes qu'il représente. Par exemple, le responsable des développeurs précise que si telle fonctionnalité est retenue, une dizaine de journées de développement est requise, à plein temps, pour un développeur. Le chef de projet, quant à lui, se fait le porte-parole du client au niveau de ses contraintes économiques et temporelles. Après discussions, les besoins sont formalisés sous forme de besoins fonctionnels, de contenu et utilisateurs, puis ils sont transmis à une partie de l'équipe technique, chargée de la première version du prototype de l'interface.

Conception basée sur prototype

Les développeurs d'interface et les graphistes conçoivent un prototype de l'interface à partir des besoins exprimés. Plusieurs itérations sont opérées sur ce prototype jusqu'à ce qu'il prenne une forme satisfaisante et validée par chaque intervenant, y compris les clients institutionnels qui participent à certaines de ces réunions de conception. À plusieurs stades, considérés comme significatifs au niveau des choix de conception, le prototype fait l'objet de tests utilisateurs, menés par des experts en utilisabilité, avec des utilisateurs finaux, réels si le public est ciblé, représentatifs si le public est vaste. Les observations réalisées lors de ces tests alimentent les discussions au sein de l'équipe de conception pour l'évolution du prototype vers le stade

final où il satisfera chacun d'eux. La technique des *Personas* (cf. §2.2.4) est employée pour personnifier les utilisateurs finaux. Cette technique s'avère efficace : nous avons observé, au cours des réunions de conception, les intervenants utiliser régulièrement les noms des *Personas* pour faire référence à un aspect particulier de l'utilisation de l'application et appuyer leur argumentation.

Les questions de contenu éditorial sont évoquées également au cours de ces itérations. Le contenu dynamique des pages, qui sera ultérieurement généré à partir d'une base de données (p.ex. la liste des entreprises d'une ville donnée) est fixé de façon arbitraire, avec des données factices mais réalistes, pour que les discussions soient fructueuses. En revanche, toutes les données fixes, connues d'avance (p.ex. les mentions légales de l'application, les éléments des menus) apparaissent sur le prototype et leur pertinence est discutée dès cette phase de prototypage. Les conflits qui apparaissent lors de ces phases de prototypage sont parfois difficiles à trancher. Lorsque plusieurs institutions sont engagées dans un projet, les questions de contenu éditorial peuvent devenir de vrais casse-tête diplomatiques : le choix du logo apparaissant en tête peut ainsi être source de discussions très longues, freinant les discussions concernant le reste du contenu, et a fortiori le reste des considérations sur l'interface. Les concepteurs expérimentés connaissent les bonnes pratiques, les choix éprouvés, mais il reste difficile de convaincre les autres intervenants, avec leurs propres intérêts, leur propre vision des choses et leur propre vocabulaire. Des pertes de temps considérables sont ainsi relevées, faute de décisions franches, faute d'argumentation suffisamment massive pour que les différentes parties s'inclinent.

Implémentation et déploiement de l'application

Sur la base du prototype d'interface validé par l'équipe de conception, l'équipe de développement implémente la partie *back-end* de l'application, c'est-à-dire l'épine dorsale, les modules fonctionnels derrière l'interface. Interviennent alors les développeurs dont les travaux sont alimentés par les réalisations du département de recherche de l'entreprise : intégration de bases de données, mise en place de processus de cryptographie, installation de protocoles de sécurité ou encore emploi de documents électroniques. Les développeurs ont le réflexe de mettre à profit leur expérience en conception chez SmalS, ou de consulter leurs collègues pour savoir si des applications similaires ont déjà été implémentées. Si c'est le cas, des fragments de code sont récupérés d'anciennes applications et intégrés à la nouvelle, bien qu'aucun cadre formel de gestion et de stockage de ces connaissances en conception ne soit à leur disposition.

Cette phase d'implémentation pure représente en moyenne 20% du temps passé sur un projet, ce qui prouve la part importante que l'entreprise accorde à la conception de l'interface et à l'utilisateur. Une fois l'application déployée, c'est encore SmalS qui est en charge de l'assistance utilisateur et de la maintenance de l'application. Une assistance téléphonique est disponible pour les utilisateurs. Lorsque des problèmes sont relevés de façon récurrente, ils sont communiqués à l'équipe du projet, pour une modification qui peut faire l'objet d'un nouveau projet si son impact sur l'application est important. Un tracé est gardé des problèmes qui font l'objet d'appel, mais aucun traitement formel n'est opéré sur ces données à ce jour et à notre connaissance.

3.2. ANALYSE DES BESOINS AU VU DES SOLUTIONS EXISTANTES

L'analyse des interfaces des applications existantes a révélé la présence récurrente d'enchaînements d'écrans et de pages. Une procédure de standardisation des interfaces serait un moyen efficace de garantir l'utilisabilité des applications d'e-Procuration, en incitant les concepteurs à réutiliser seulement les bonnes pratiques de conception (Pontico, Winckler et Limbourg 2007b). La standardisation garantirait des solutions prêtes à l'emploi et utilisables, plus sûrement que par le biais des recommandations ergonomiques telles que celles actuellement fournies par les organismes de législation en matière d'e-Gouvernement. Un gain de temps pour les entreprises s'ajouterait à des garanties de réussite en matière d'utilisabilité.

La standardisation garantirait aussi à l'utilisateur une cohérence transversale au travers des applications jouant le jeu de la standardisation. L'avantage pour l'utilisateur tient alors dans un apprentissage du mode de fonctionnement des principes interactifs de ces applications une fois pour toutes. Lorsqu'il visite une première fois une application, il apprend à l'utiliser, découvrant la façon dont il peut interagir avec elle. Ensuite, qu'il visite une deuxième fois cette application, ou qu'il en visite une nouvelle, la standardisation lui permet de déjà connaître les modes d'interaction, et rend sa visite plus efficace et agréable.

L'usage de patrons d'interfaces permettrait d'offrir ces solutions sous forme générique, de spécifier le contexte d'usage d'un patron et de renvoyer éventuellement le concepteur vers un patron plus adapté à son contexte, et de rassembler les bonnes pratiques pour assurer l'utilisabilité de l'application. L'aspect récurrent des applications d'e-Gouvernement a déjà été exploité pour les processus, par le biais de patrons de processus (Verginadis, Gouscos, et Mentzas 2004), issus de courants de réflexion sur la modélisation du travail inter organisationnel.

Cependant, au vu de notre état de l'art de la littérature (Pontico, Winckler et Limbourg 2007a), il n'a pas été publié à l'heure actuelle de catalogue de patrons d'interface pour l'e-Gouvernement. Cette absence est sans doute due au fait que la maturité requise pour l'établissement de patrons a été atteinte très récemment dans le domaine : dégager de bonnes conduites de conception nécessite en effet un recul important sur les alternatives de conception et leurs conséquences comparées. En revanche, de nombreux patrons ont été définis pour l'e-Commerce (p.ex. (Martijn Van Welie 2007)), domaine proche de l'e-Procuration sur certains aspects (Wimmer 2001). Cependant, la nature différente de ces deux domaines, quoiqu'ils aient l'aspect procédural en commun, fait émerger bien vite de capitales différences du point de vue de l'interface. Un patron d'interface pour l'e-Commerce inclura des éléments d'incitation à l'achat, tandis qu'un patron de page d'e-Procuration, loin de songer à susciter de nouveaux besoins chez l'utilisateur, proposera une réponse formelle et claire aux seuls besoins qui l'ont poussé à visiter l'application. De plus, la définition de patrons doit venir du contexte d'utilisation, de l'expérience de conception du domaine, d'observations acérées sur les conditions de conception : cela rend bancale l'utilisation pour l'e-Procuration de patrons prévus pour d'autres domaines. Nous discutons dans cette section de l'intérêt qu'apporterait un catalogue de patrons d'interfaces spécifiques à l'e-Procuration dans la conception d'applications de ce domaine.

3.2.1. Patrons d'interface pour l'e-Procuration

L'intégration d'un nouveau support à la conception tel qu'un catalogue de patrons d'interfaces doit se baser sur les activités de conception existantes : en cas de conception basée sur des prototypes, il est ainsi pertinent d'inclure des prototypes basse fidélité à des patrons d'interface (Martjin Van Welie 2002) afin d'ancrer l'utilisation des patrons comme étape concrète dans le processus de conception. Le contenu du catalogue de patrons doit être conforme aux besoins de ses utilisateurs, proposer des solutions à la fois non ambiguës et faciles à lire et modifier, illustrées par de bons et mauvais exemples. Lorsque la solution proposée par le patron provient d'un raisonnement scientifique (par opposition à un patron issu d'observations empiriques pures), la solution sera étayée par des références scientifiques. L'utilisateur s'y réfèrera pour approfondir sa compréhension du patron, proposer des modifications pour ce patron ou encore proposer un patron complémentaire. Cet aspect proactif étant à favoriser, tant comme facteur d'adoption de ses utilisateurs que comme garant de la pertinence des patrons (Fincher et Windsor 2000), un catalogue de patrons est sujet à de nombreuses évolutions. L'aspect évolutif et la perspective d'un gros volume de patrons à traiter doivent ainsi être considérés lors de l'établissement des moyens d'organisation et de consultation des patrons.

Un catalogue de patrons présenté sous forme d'une application hypermédia (p.ex. un catalogue sous forme d'application Web tel que (Martijn Van Welie 2007)) favorise l'interaction avec ses utilisateurs : un système de commentaires est mis en place, et les modifications sont peu coûteuses. Au contraire, un support écrit (p.ex. le livre de (Van Duyne, Landay, et Hong

2002)) est prisé par des utilisateurs rétifs à la lecture sur écran, mais présente une grande rigidité quant à son évolutivité. Toutefois, quelque soit le support choisi, des modes de fonctionnement communs pour l'organisation des patrons émergent : un patron voisin peut ainsi être référencé par un lien hypertexte ou un renvoi vers la page le décrivant. L'organisation des catalogues de patrons étudiés suit souvent le fil d'un raisonnement de conception classique, allant de problèmes généraux à des problèmes plus précis (p.ex. le catalogue (Montero et coll. 2002) enchaîne les patrons de conception de sites Web suivant une granularité croissante, du « niveau site Web » au « niveau ornementation »).

Les problématiques d'organisation sont un thème de recherche crucial et encore en cours d'affinement dans la communauté des patrons de conception en général, au-delà des patrons de conception d'interface. L'objectif est de fournir une organisation proposant un accès rapide et facile au patron approprié au problème qui se pose à l'utilisateur des patrons. Sally Fincher (Fincher et Windsor 2000) établit les critères d'organisation suivants : facilité de localisation, groupement de patrons apparaissant dans des cas d'utilisation communs, offre de différents points de vues, incitation à générer de nouvelles solutions à partir de celles proposées. Plusieurs auteurs tentent de fournir une organisation efficace en établissant des réseaux entre patrons de conception, des liens entre eux qui soient riches en sémantique. James Noble (Noble 1998) propose de lier les patrons de conception entre eux par des relations primaires (utilise, affine, est en contradiction avec) et des relations secondaires (utilisé par, affiné par, variante de (dans le problème ou dans la solution exposés par le patron), variante de tel patron utilise tel autre, est similaire à, combine, requiert, s'utilise lui-même, entre dans une séquence de patrons). Martijn Van Welie (Martijn Van Welie et Van der Veer 2003) propose quant à lui une organisation inspirée de la modélisation orientée objet, ce qui permet de décrire les patrons sous forme de hiérarchie, et d'établir entre certains d'entre eux des relations d'agrégation, de composition, de spécialisation ou d'association simple.

Scott Henninger (Henninger et Ashokkumar 2005) suggère que l'organisation de patrons utilise les outils fournis par Web sémantique : associer à chaque patron un ensemble de métadonnées enrichit la sémantique des relations pouvant être tissées entre patrons. La métadonnée « contexte d'utilisation du patron » pourrait ainsi permettre de rassembler les patrons apparaissant dans des cas d'utilisation communs, comme le conseille Sally Fincher. De plus, le domaine du Web sémantique a pour objectif, à terme, l'indexation du Web dans son ensemble, ce qui sous-entend la prise en compte d'un volume d'informations à la fois important et fortement évolutif. Chaque élément du réseau est responsable des données qu'il fournit au réseau : il renseigne ses propres méta données et s'intègre alors dans la structure sémantique fournie, par exemple une ontologie.

3.2.2. Organisation des patrons pour la conception en e-Gouvernement

Le Tableau 4 ci-après compare les différents modes d'organisations évoqués dans cet état de l'art en fonction de critères basés sur ceux de Sally Fincher (groupement d'éléments proches et comparaison de points de vue) et sur nos propres observations (références croisées, structure hiérarchique et évolutivité). Nous considérons ainsi que pour une organisation efficace favorisant la recherche de solutions de conception adéquates, un catalogue de patrons devrait présenter les avantages suivants :

- **Structure hiérarchique :** l'utilisateur doit pouvoir naviguer au fil des patrons suivant une réflexion de conception classique, partant de problèmes de conception de niveau abstrait pour aller vers des problèmes plus concrets ;
- **Références croisées entre éléments d'interface :** l'utilisateur doit pouvoir obtenir l'ensemble des patrons concernant un élément d'interface particulier (p.ex. tous les patrons concernant les formulaires) ;
- **Groupement d'éléments proches :** les patrons apparaissant souvent dans des contextes d'utilisation identiques ou similaires doivent être groupés ;

- **Comparaison de points de vue :** dans certains cas, plusieurs patrons semblent adéquats. Ils doivent être groupés, et l'utilisateur doit disposer des informations qui lui permettront de faire un choix, d'après son contexte de conception ;
- **Évolutivité et passage à l'échelle :** le catalogue de patrons doit supporter avec souplesse des ajouts, des modifications, des commentaires et des suppressions de patrons. Le catalogue doit supporter également d'atteindre un volume d'informations important sans rien perdre de ses atouts organisationnels. Nous évaluons ici l'organisation, sans tenir compte du support qui fait éventuellement obstacle à cette évolution.

Tableau 4. Comparaison des méthodes d'organisation des catalogues de patrons d'interfaces étudiés

	Structure hiérarchique	Références croisées entre éléments d'interface	Groupement d'éléments proches	Comparaison de points de vue	Évolutivité et passage à l'échelle
(Borchers 2000)	✓	✗	✓	✓	~
(Coram et Jim Lee 2002)	✓	✗	✓	✓	✗
(Henninger et Ashokkumar 2005)	✓	✓	✓	✓	✓
(Laakso 2003)	✗	✗	✗	✗	✗
(Montero et coll. 2002)	✓	✗	✗	✗	✗
(Noble 1998)	✓	✗	✓	✓	~
(Perzel et Kane 1999)	~	✗	✗	✗	✗
(Tidwell 2005)	✓	✗	✓	✓	~
(Van Duyne, Landay, et Hong 2002)	✓	✗	✓	✓	~
(Martijn Van Welie 2007) (catalogue)	✓	✗	✓	✓	✗
(Martjin Van Welie et Van der Veer 2003) (orientation objet)	✓	✗	✓	✓	~
(Yahoo! 2007)	~	✗	✓	✓	✗

LÉGENDE ✓ supporté, ✗ non supporté, ~ partiellement supporté

Cette comparaison donne clairement l'avantage à une organisation basée sur des principes issus du Web sémantique telle que le préconise Scott Henninger (Henninger et Ashokkumar 2005). Un outil est associé à cette méthode, afin d'aider à l'édition d'ontologie pour le tissage des patrons : il s'agit de BORE (*Building an Organizational Repository of Experiences*[1]), qui n'est malheureusement plus disponible à l'heure actuelle[2]. De puissants outils de visualisation d'ontologies existent toutefois et peuvent a fortiori servir pour une ontologie de patrons : citons parmi eux Jambalaya (Storey et coll. 2001), un outil de visualisation greffé à l'environnement d'édition d'ontologies Protégé (Stanford Center for Biomedical Informatics Research 2008).

3.3. Analyse des interfaces des applications développées

Lors du séjour chez SmalS, nous avons eu l'opportunité, en plus de l'observation des activités de conception, de participer à la mise en œuvre du StyleGuide, par le biais de l'étude théorique présentée ci-dessus (cf. §3.2), mais aussi par une analyse des applications existantes. Nous y avons observé a priori une certaine récurrence dans les interfaces d'applications d'e-Procuration, corroborée par les réflexes de réutilisation des développeurs de l'entreprise (cf. §3.1.2). Nous avons souhaité confirmer ou infirmer cette impression par une étude statistique. Cette étude a été menée sur quinze applications développées par SmalS, et toutes liées au même

[1] Construire un dépôt organisé d'expériences
[2] BORE était disponible à l'adresse http://cse-ferg41.unl.edu/bore.html (dernier accès : 25 sept. 08)

organisme national de la sécurité sociale belge. Ce contexte précis favorise la réutilisation de composants de conception, puisqu'une même culture d'entreprise est partagée, et que le contexte de déploiement est le même. Nous y avons effectivement relevé des éléments d'interface récurrents, de différents niveaux de granularité, selon les niveaux indiqués sur la pyramide de la Figure 26. Les éléments des deux plus hauts niveaux (enchaînements d'écrans et pages) sont listés et analysés dans cette section.

- **Enchaînement d'écrans :** comportement de séquences de pages, correspondant à un comportement fonctionnel de l'interface (p.ex. assistant à la réalisation d'un formulaire en plusieurs étapes) ;
- **Page :** mise en page d'une page ou d'un élément inclus dans une page, mais significatif par lui-même de l'interface utilisateur (p.ex. formulaire) ;
- **Composant de base :** affichage et offre d'interactions sur des éléments d'interface qui n'ont pas de sens par eux-mêmes, mais qui apparaissent dans des éléments de plus haut niveau (p.ex. champ de formulaire).

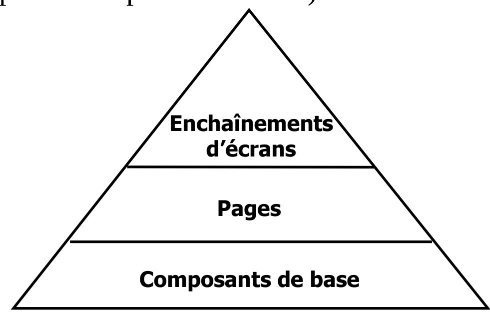


Figure 26. Niveaux de granularité des fragments d'interface rencontrés en e-Gouvernemente-Procuration

3.3.1. Enchaînements d'écrans récurrents

Les applications d'e-Procuration procédurales sont différenciables par leur comportement en termes de navigation. Chacune présente en effet un comportement particulier en terme de structure hypertexte et donc d'enchaînements d'écrans. Ces comportements de navigation sont adaptés à des types d'actions et d'utilisateurs particuliers, ce qui autorise la combinaison de plusieurs de ces enchaînements d'écrans si une même application offre plusieurs types de fonctionnalités. Plusieurs familles d'applications se distinguent caractérisées par un enchaînement d'écrans similaire. L'Annexe A décrit chacune de ces familles, listées ci-dessous, en s'appuyant sur un exemple relevé lors de notre analyse de l'existant. Une description formelle de ces enchaînements d'écrans est proposée également, utilisant le formalisme de modélisation de la navigation Web StateWebCharts[1] (Winckler et Palanque 2003).

- Un **assistant** est une séquence de pages qui guide l'utilisateur au travers des étapes de réalisation d'une procédure (p.ex. déclaration de revenus) ;
- Une application de **consultation et modification de données** autorise l'utilisateur à consulter et modifier un ou plusieurs des éléments diffusés par le biais d'une liste (p.ex. consultation des entreprises ayant déclaré une période de chômage temporaire) ;
- La **gestion de dossiers** permet la conduite de plusieurs activités en parallèle, sur des éléments de nature similaire, à différents niveaux d'avancement (p.ex. ajout et suivi de dossiers concernant des réfugiés politiques) ;

[1] Seule une partie du pouvoir expressif de la notation SWC sera utilisée ici, à savoir la partie concernant les enchaînements d'écran, visible par l'utilisateur. La partie concernant les appels au serveur seront laissés de côté pour se concentrer uniquement sur les interactions avec l'utilisateur. Il est toutefois indiqué que le contenu d'une page est statique (rectangle aux bords arrondis, en trait plein) ou généré dynamiquement (rectangle aux bords arrondis, en pointillés).

- La **gestion de rôles** apparaît lorsqu'une même application sert à plusieurs catégories d'utilisateurs, chacune étant associée à un ensemble de tâches autorisé (p.ex. dépôt et consultation d'offres d'emploi) ;
- Une **architecture *"hub and spoke"***, littéralement « centre et rayon », propose plusieurs tâches à l'utilisateur, chacune d'elles ayant pour point de départ une page faisant office de tableau de bord. Les tâches sont réalisées les unes après les autres, et à la fin de chaque tâche, l'utilisateur est renvoyé à la page de tableau de bord (p.ex. gestion des informations concernant les employés d'une entreprise) ;
- L'**intégration à un portail** concerne le référencement d'une application depuis un portail (p.ex. applications liées au portail de la sécurité sociale).

3.3.2. Pages récurrentes

Les enchaînements d'écrans que nous avons observé de façon récurrente sont composés de pages qui apparaissent a fortiori de façon récurrente sur les applications d'e-Procuration. L'Annexe A les décrit en donnant pour chacune d'elles un exemple issu d'applications de SmalS. Les captures d'écran ont été réalisées en octobre 2006, durant le séjour sur le terrain, dans le but de collecter des exemples de conception de différents niveaux de qualité. Ces captures d'écran montrent des applications telles qu'elles étaient à l'époque, en déploiement ou en phase de test en interne. Une fois les différents comportements de conception collectés, les bonnes attitudes à standardiser pouvaient ainsi émerger de cette étude. Plusieurs catégories de pages coexistent, selon le rôle qu'elles jouent dans l'interaction avec l'utilisateur :

- Diffusion de données
  - Un **accusé de réception** est affiché lorsqu'une procédure a été accomplie avec succès et validée par l'utilisateur ;
  - Une **liste d'informations à rassembler** est affichée en début de procédure, pour la bonne conduite de la procédure ;
  - Des **données contextuelles** sont affichées sur la base de l'historique de la navigation ou bien par rapport au compte de l'utilisateur, pour offrir des repères ;
  - Un **récapitulatif** est affiché lorsque toutes les étapes d'une procédure ont été accomplies, donnant la possibilité à l'utilisateur de confirmer ou d'annuler la procédure ;
  - Une **liste d'éléments** est affichée généralement sous forme de tableau, et offre des modes de tri facilitant sa consultation ;
  - Une **liste de tâches** permet un guidage de l'utilisateur en fonction des tâches qu'il peut réaliser avec l'application ;
  - Des **données d'horodatage** identifient de façon unique la tâche en cours de réalisation ; l'utilisateur peut ainsi prouver qu'il a souhaité réaliser la procédure, ou encore obtenir une aide efficace sur l'utilisation de l'application ;
- Offre de navigation
  - Les **points d'entrée directs** offrent un accès à une application au travers du choix d'une tâche, d'une langue ou d'une catégorie d'utilisateurs, choix réalisé parmi peu d'éléments ;
  - Une **page de tableau de bord** offre un accès à plusieurs tâches depuis une même page, au sein d'une architecture "*hub and spoke*" ;
  - Un **filtrage de liste** est proposé pour masquer les éléments n'ayant pas d'intérêt pour la tâche en cours ;
  - Une **recherche** d'élément est réalisée sur l'ensemble des éléments stockés dans la base relative à une application ;

- La **navigation de service** apparaît au fil des pages de l'application, généralement en tête de page, pour que les liens principaux soient toujours à disposition de l'utilisateur ;
- Un **tri de liste** permet le classement d'éléments pour une recherche ou une sélection efficace ;

- Entrées de l'utilisateur
  - Un **formulaire de contact** permet à l'utilisateur de transmettre une remarque ou une question à l'équipe de maintenance de l'application ou directement à l'institution associée à l'application ;
  - Un **formulaire de déclaration** offre la réalisation d'une tâche en une seule page ;
  - Une **page d'identification** permet à l'utilisateur de s'identifier pour accéder aux tâches ou informations que l'application lui propose ;
  - Une **étape d'assistant** fait partie d'une procédure à réaliser en plusieurs étapes ;
- Un **guidage** est offert pour le remplissage de formulaire, donnant des informations sur les formats de données attendus et les erreurs de saisie.

3.3.3. Synthèse

Les concepteurs de SmalS, même non pourvus de moyens formels de gestion de la compétence en matière de conception, capitalisent les efforts fournis sur d'anciens projets pour les projets en cours : des fragments d'interfaces identiques sont ainsi rencontrés au fil des applications. Certains sont plus pertinents que d'autres du point de vue de l'utilisabilité de l'application ; certaines inclusions sont notamment de mauvais aloi du point de vue graphique.

Le Les statistiques présentées dans ce tableau montrent que nombre d'éléments d'interface apparaissent effectivement de façon récursive, comme nous l'avions pressenti. De plus, la plupart d'entre eux sont déjà standardisés : l'élément page « Recherche » apparaît toujours sous la même forme, seuls les critères de recherche changent, selon le contexte. D'autres éléments apparaissent sous plusieurs formes, mais ces formes sont souvent plusieurs standards à utiliser dans des contextes de conception différents (p.ex. l'élément page « Page d'identification » apparaît sous deux formes : l'une est adaptée à l'identification d'un citoyen, l'autre à l'identification d'un agent administratif).

Tableau 5 présente les enchaînements d’écrans et les pages rencontrés de façon récurrente lors de l’analyse de l’interface de quinze applications développées par SmalS. Leur fréquence d’apparition apparaît sous forme de pourcentage : une fréquence d’apparition de 100% correspond à un élément d’interface qui est apparu sur les quinze applications. Le nombre de formes distinctes que prend cet élément de l’interface au fil de ses apparitions est indiqué également. Le chiffre 1 correspond à un élément qui apparaît toujours sous la même forme, au travers des applications. Lorsque l’expression « >3 » apparaît, l’élément est apparu sous plus de trois formes distinctes. Lorsque deux formes sont proches en termes d’éléments d’interface les composant, elles sont considérées comme une seule forme. Deux formes de composition principale similaire, mais offrant des différences en termes de couleurs, d’intitulés, ou encore de mise en page secondaire, seront ainsi considérés comme de la même forme.

Les statistiques présentées dans ce tableau montrent que nombre d’éléments d’interface apparaissent effectivement de façon récursive, comme nous l’avions pressenti. De plus, la plupart d’entre eux sont déjà standardisés : l’élément page « Recherche » apparaît toujours sous la même forme, seuls les critères de recherche changent, selon le contexte. D’autres éléments apparaissent sous plusieurs formes, mais ces formes sont souvent plusieurs standards à utiliser dans des contextes de conception différents (p.ex. l’élément page « Page d’identification » apparaît sous deux formes : l’une est adaptée à l’identification d’un citoyen, l’autre à l’identification d’un agent administratif).

Tableau 5. Statistiques sur la présence récurrente d'éléments d'interfaces sur des applications d'e-Procuration

Élément d'interface			Fréquence d'apparition	Nombre de formes distinctes
Enchaînements d'écrans		Consultation et modification de données	33%	3
		Gestion de dossiers	33%	3
		Architecture *"hub and spoke"*	26%	>3
		Intégration à un portail	100%	1
		Gestion de rôles	13%	2
		Assistant	80%	3
Pages	**Diffusion de données**	Accusé de réception	60%	1
		Liste d'informations à rassembler	20%	1
		Données contextuelles	100%	>3
		Récapitulatif	86%	1
		Liste d'éléments	86%	>3
		Liste de tâches à réaliser	20%	2
		Données d'horodatage	60%	1
	Offre de navigation	Points d'entrée directs	33%	>3
		Page de tableau de bord	33%	>3
		Filtrage de liste	46%	>3
		Recherche	80%	1
		Navigation de service	100%	>3
		Tri de liste	73%	1
	Entrées de l'utilisateur	Formulaire de contact	100%	1
		Formulaire de déclaration	80%	1
		Page d'identification	100%	2
		Étape d'assistant	80%	>3
	Guidage	Erreurs sur les formulaires	100%	>3

3.4. Conclusion

L'observation exploratoire de SmalS que nous venons de décrire dans ce chapitre 3 a permis de mettre en avant la pertinence d'un guidage de la conception d'interfaces d'e-Procuration basé sur des patrons. Suite à notre séjour d'octobre 2006, SmalS a opérationnalisé le StyleGuide. Un comité de pilotage du StyleGuide a été mis en place, ayant pour rôle de valider le contenu du StyleGuide à sa conception, et au fil de ses évolutions. Des formes Visio génériques liées aux patrons de pages et de composants de base ont été créées pour soutenir les analystes dans la conception de *wireframes*. Des fragments de code XHTML, JavaScript et CSS ont été intégrés aux patrons à destination des développeurs d'interfaces. Fin 2007, le StyleGuide était utilisé sur des projets pilote. Peu à peu, une trentaine de personnes a utilisé le StyleGuide, sur des projets pilote et sur des projets classiques. L'été 2008 verra paraître une version du StyleGuide stabilisée dans sa forme.

En avril 2008, alors que le StyleGuide est utilisé par une trentaine de personnes, analystes et développeurs d'interfaces en particulier, nous avons eu l'opportunité de retourner sur place durant une semaine pour mener une évaluation. La rencontre et l'observation de faits provoqués avec des analystes utilisateurs du StyleGuide nous a permis d'en observer l'utilisabilité, l'acceptation, l'adéquation aux activités de conception. La mise en place et les résultats de cette évaluation seront présentés dans le chapitre 5.

De notre côté, la récurrence d'éléments d'interface et l'intérêt d'une standardisation des interfaces d'applications d'e-Procuration, nous ont poussés à approfondir et formaliser une méthode de conception basée sur des patrons d'interfaces pour l'e-Procuration. Le chapitre 4 pose les fondations de cette méthode et lui donne un cadre de développement. Notre approche étant basée sur celle du StyleGuide, les évaluations menées sur le StyleGuide nous aideront à valider les partis pris de notre propre méthode et à suggérer des évolutions.

4. Méthode de conception basée sur des patrons

Résumé

Face à des problèmes de conception d'interface récurrents, les concepteurs d'e-Gouvernement réutilisent naturellement des fragments de projets passés. Notre méthode vise à formaliser cette habitude pour la rendre plus optimale et favoriser ainsi l'utilisabilité des applications conçues. Des patrons d'interfaces ont ainsi été organisés selon leur niveau de granularité et richement articulés par le biais d'une ontologie de leurs concepts. Pour que notre méthode puisse être intégrée à un cycle de conception basé sur le prototypage moyenne fidélité, un support à ce prototypage est intégré aux patrons, à destination des analystes, en tout début de conception.

Une application d'e-Gouvernement doit être utilisable par le plus grand nombre, ce qui fait de la conception de son interface une activité délicate. L'observation exploratoire des activités de conception en e-Procuration (cf. chapitre 3) a révélé la forte récursivité des problèmes de conception d'interfaces à traiter dans les projets successifs. Instinctivement, les concepteurs d'interface, de l'analyste au développeur, récupèrent des fragments de projets passés pour gagner du temps et puiser des idées. Des fragments d'interface reviennent ainsi régulièrement dans les applications produites par une même entreprise, comme nous l'avons observé précédemment (cf. §3.2). Formaliser ce réflexe de réutilisation permettrait de rendre cette habitude de conception plus :

- **Efficace :** gain de temps dans la recherche du fragment recherché au sein des projets déjà développés, mise à disposition pour les nouveaux entrants dans l'entreprise ayant peu de recul sur les projets passés, conception de fragments génériques pour un gain de temps lors de l'adaptation au projet courant ;
- **Valide :** stockage de fragments validés par un comité central garantissant la qualité du fragment et spécifiant ses cas d'utilisation ;
- **Riche :** partage d'expériences gratifiant, incitation des concepteurs à contribuer à cette base de solutions, diffusion de connaissances en utilisabilité.

La méthode que nous proposons vise ces objectifs, et se base pour cela sur des patrons de conception d'interfaces : ils servent de support à l'analyse, en tout début de conception d'une application d'e-Procuration (cf. §4.1). Tous les patrons contiennent bons et mauvais exemples et des recommandations de mise en place. Notre méthode se positionne comme support au prototypage moyenne fidélité de l'interface, enchaînements d'écrans, pages et composants de base. Les prototypes de patrons d'enchaînement d'écrans sont formalisés au moyen d'un langage de modélisation de la navigation Web et les patrons de pages et de composants de base sont accompagnés de *wireframes*. Ces patrons génériques sont à adapter au contexte de conception en cours. Les observations réalisées dans le chapitre précédent (cf. §3.2) ont guidé nos choix pour une présentation optimale de ces conseils de conception.

Les patrons sont regroupés dans un catalogue, un site Web qui encourage la navigation d'un patron à l'autre (cf. §4.2). Pour faciliter la lecture et aider l'utilisateur à se repérer, chaque patron est décrit systématiquement selon un ensemble d'attributs. Les patrons sont organisés selon leur niveau de granularité, pour que l'ordre de réflexion naturel de la conception soit respecté. Cette organisation est complétée par une ontologie des concepts inclus dans les patrons.

Un catalogue de patrons étant par nature évolutif, une procédure de suggestion d'ajout, de la part des utilisateurs du catalogue est décrite en association avec notre méthode (cf. §4.3). Lors d'éventuelles mises à jour du catalogue, des recommandations en terme de communication sont

délivrées. La bonne acceptabilité d'un support fortement évolutif tel qu'un catalogue de patron nécessite en effet une excellente visibilité des changements qui lui sont apportés. Deux scénarios d'utilisation de la méthode illustrent ce chapitre (cf. §4.4) : l'un d'eux intègre un outil de support qui sera présenté dans le chapitre 6.

4.1. CYCLE DE DÉVELOPPEMENT INTÉGRANT LA MÉTHODE

Le cycle de vie recommandé pour l'intégration de notre méthode de support à la conception d'interfaces pour l'e-Gouvernement est basé sur le cycle de vie en O (Scapin et coll. 2000). Nous avions noté dans l'état de l'art que le cycle de vie en O pouvait intégrer des méthodes de prototypage et qu'il présentait des atouts en termes de gestion de l'évolutivité de l'application (cf. §2.1.8). Les étapes de ce cycle de vie sont ici adaptées à l'e-Gouvernement et aux méthodes que nous proposons comme support. La Figure 27 montre ces adaptations : les outils utilisés pour une étape sont indiqués à côté de celle-ci, en lettres capitales ; les documents produits par une étape sont indiqués sur la transition sortant de cette étape.

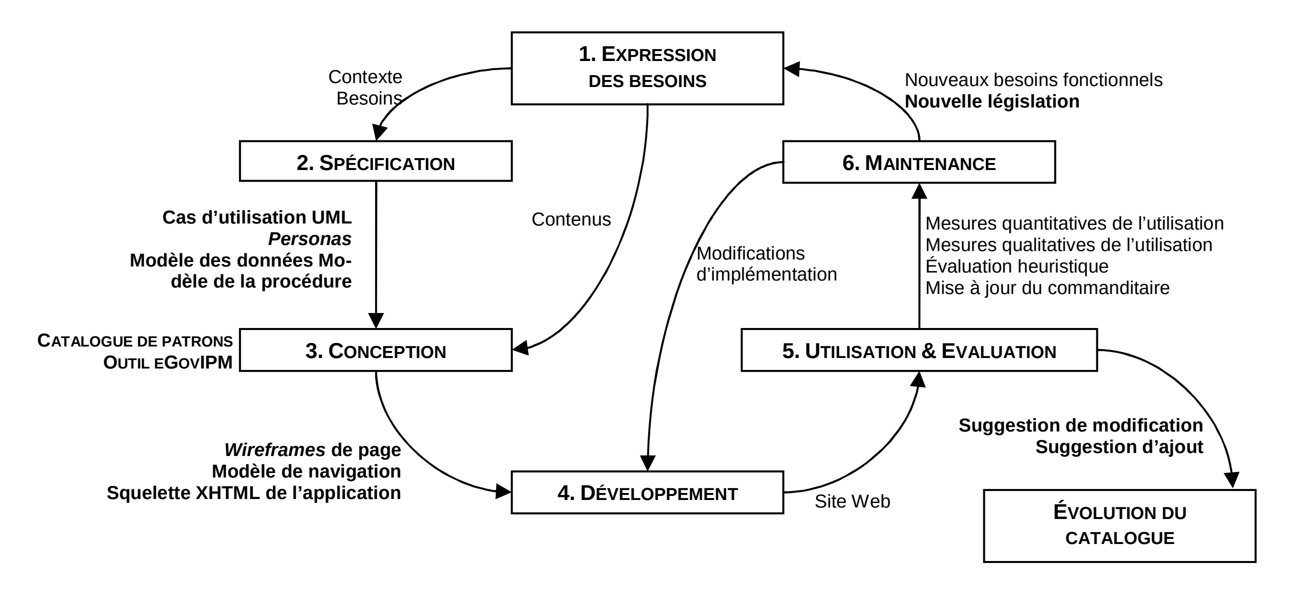


Figure 27. Cycle de développement pour l'e-Gouvernement, basé sur des patrons d'interface

Cette section offre des recommandations sur le déroulement des phases pour lesquelles, en l'état, notre méthode offre une contribution : la Spécification, la Conception et le Développement. La phase d'évolution du catalogue, qui fait partie du cycle de vie de l'application mais pas de son cycle de développement, sera traitée plus loin dans le chapitre (cf. §4.3).

4.1.1. Expression des besoins et Spécification

Le recueil des besoins doit se faire auprès du client, c'est-à-dire auprès du point de contact de l'institution demandeuse de l'application à développer. La discussion est orientée par l'analyste pour obtenir suffisamment d'informations et pouvoir concevoir les documents de formalisation des besoins. Faute de temps, cette discussion est parfois remplacée par un bref descriptif des besoins par le client ; des itérations de redéfinition des besoins étant alors à prévoir, la première définition de l'interface n'est pas approfondie.

La formalisation des besoins exige une couverture large des fonctionnalités de l'application par l'analyste. Il est suggéré tout d'abord de lister les fonctionnalités par le biais des fonctionnalités accessibles via l'application à concevoir. Pour ce faire, un diagramme de **cas d'utilisations** UML est adapté : les acteurs de l'application sont associés à des tâches que l'application leur donne la possibilité de réaliser. Lorsque la catégorisation de l'acteur modifie les fonctionnalités qui lui sont accessibles, et que cette catégorisation passe par une action de sa part (p.ex. saisie d'un identifiant et d'un mot de passe), l'acteur « Visiteur non identifié » sera

associé à cette fonctionnalité de reconnaissance ; les profils identifiables figureront à ses côtés, associés aux fonctionnalités qui leur sont réservées.

Les acteurs de l'application sont ensuite décrits sous un angle différent, celui de l'illustration, avec la technique des ***Personas*** (cf. §2.2.4). Une *Persona* au moins est décrite, appartenant au profil prévu comme étant le plus fréquent visiteur de l'application. Ses caractéristiques et objectifs dépassent la façon dont il est vu sur cette application : ses habitudes vis-à-vis de l'utilisation du Web, ses compétences en informatique, son degré d'utilisation de l'e-Gouvernement, son attitude vis-à-vis des nouvelles technologies seront notamment décrits.

Même dans les entreprises pourvues de responsables éditoriaux et d'experts en bases de données, l'analyste est le premier à suggérer un modèle de données. Cette étape est nécessaire pour une description cohérente de l'interface, notamment de l'information mise à disposition, des fonctionnalités autorisées sur ces informations, de la gestion des profils sur l'application. Il est ici suggéré de réaliser ce modèle de données préliminaire sous forme d'un modèle **Entité – Relation**. Cette notation permettra notamment de représenter les actions que chaque profil peut réaliser sur chaque donnée, par le biais de relations entre entités de profils et de données.

Dans le cas d'applications d'e-Procuration procédurales, la procédure est modélisée par le biais d'un formalisme adapté, par exemple avec un **diagramme d'activités**. Les interactions entre agents et système sont ainsi représentées, ainsi que les interactions avec d'éventuels systèmes annexes.

4.1.2. Conception

La phase précédente (cf. §4.1.1) a spécifié les besoins de l'application relativement aux utilisateurs, à la procédure à dématérialiser, et aux données à manipuler. Dans la présente phase de Conception, ces besoins sont exprimés en termes de conception d'interface : pages et structure hypertexte sont considérés. Notre méthode fournit un guidage pour ces activités de conception d'interface. La conception de l'interface commence avec la définition de la navigation, et continue avec la définition des pages apparaissant dans le modèle de navigation. Toutefois, une fois cette première itération franchie, les deux activités s'entremêlent. Il est conseillé de garder un modèle de navigation toujours valable, et de le mettre à jour dès que nécessaire. Il s'agit du squelette de l'application, du fil entre les pages : il pose le cadre général et permet de conserver une vue d'ensemble. D'où l'intérêt de le tenir rigoureusement à jour, pour qu'il reflète la vision courante de l'application.

Définition de la navigation

Le catalogue de patrons d'interface[1] dispose d'une section « Enchaînement d'écrans », qui présente les grandes familles d'applications que l'étude de l'existant avait révélées. L'analyste parcourt cette section à la recherche de l'enchaînement d'écrans StateWebCharts qui correspond à la nature du projet auquel il est confronté. Si l'application est complexe (p.ex. un portail), plusieurs enchaînements d'écrans doivent être combinés pour représenter la structure de navigation de l'application. Ensuite, il personnalise l'enchaînement d'écrans générique, en spécifiant par exemple le nombre d'étapes d'une procédure ou en ajoutant un page spécifique : l'outil eGovIPM supporte cette activité en contrôlant que la syntaxe de l'enchaînement d'écrans est toujours correct (cf. chapitre 6). L'outil eGovIPM génère ensuite un ensemble de pages au format XHTML, liées entre elles conformément au modèle personnalisé par l'analyste. Sur chaque page apparaît son titre, une description succincte, son type (correspondant à une entrée dans le catalogue de patrons), les liens sortants et leurs déclencheurs, ainsi qu'un *wireframe* générique du type de page concerné. Ce *wireframe* générique peut par la suite être remplacé par le *wireframe* de page conçu par l'analyste.

[1] http://fpontico.free.fr/ (dernier accès : 25 sept. 08)

Définition de wireframes *de page*

Les pages apparaissant dans la structure hypertexte doivent maintenant être décrites plus en détails. Les *wireframes* réalisés par les analystes servent à mettre en avant les fonctionnalités de l'application : il n'est donc pas question, ici, d'identité graphique. L'analyste balaie à présent les sections « Page » puis « Composant de base » du catalogue de patrons d'interface[1]. Il récupère les pages identifiées depuis l'enchaînement d'écrans adapté, et éventuellement des pages supplémentaires selon ses besoins. Puis il imbrique dans ces pages, aux endroits laissés vides par le *wireframe* générique, des informations ou des formulaires ayant trait au projet courant. Il est conseillé de réaliser cette activité avec un outil d'édition de schémas de type Microsoft Visio. D'autres endroits, laissés vides par les *wireframes* génériques du catalogue, seront comblés avec des *wireframes* de composants de base, selon les besoins de l'application. L'analyste enregistre les *wireframes* de page au format image, afin de pouvoir les faire apparaître dans l'enchaînement d'écrans défini avec eGovIPM.

Génération d'un squelette XHTML

L'outil eGovIPM est décrit dans le chapitre 6 : il guide l'édition de modèles de navigation StateWebCharts sur la base de patrons génériques d'enchaînement d'écrans. Une fois le modèle de navigation réalisé, eGovIPM génère le squelette de l'application au format XHTML, c'est-à-dire un ensemble de pages HTML liées entre elles selon le modèle de navigation édité. Les pages contiennent le *wireframe* générique de la page considérée ou bien un *wireframe* personnalisé si l'utilisateur d'eGovIPM l'a spécifié.

Validation avec le client

Quand l'analyste considère que le modèle de navigation et les *wireframes* de page satisfont les besoins qui ont été recueillis, il les présente au client, pour obtenir sa validation. L'enchaînement d'écrans est présenté au client, chaque page présentant le *wireframe* personnalisé pour la page, et les liens sortants, associés à des éléments de la page. Le client discute de cette solution et demande éventuellement à ce que des modifications y soient apportées. Les modifications peuvent être profondes dans le cas où, face à la proposition de l'analyste, le client réalise que des besoins ont été omis ou mal exprimés. Toute demande de modification implique un retour, pour l'analyste, à l'étape de proposition d'interface. Si aucune modification n'est demandée, et que le client est satisfait, la spécification fonctionnelle est terminée : elle est suivie de l'étape de maquettage interactif en XHTML, sur la base du squelette généré par l'outil eGovIPM, du modèle de navigation et des *wireframes* de page.

4.1.3. Développement

Ajouter des patrons de nature différente au catalogue permettrait de couvrir d'autres phases du processus de conception. En l'occurrence, des patrons de composants de base intégrant des fragments de code (comme fourni par le StyleGuide cf. §3.4) permettraient de fournir un support concret à la phase de Développement.

En l'état actuel d'avancement de notre méthode, la contribution à la phase de développement tient dans le support à la communication entre analystes et développeurs : le squelette de l'application généré automatiquement par l'outil eGovIPM à partir du modèle de la structure hypertexte et des *wireframes* personnalisés par les analystes. Ce squelette de l'application offre une définition des modalités de navigation, et de la disposition des composants de base dans les pages.

Les aspects graphiques sont discutés et décidés lors du développement, par des graphistes qui ont une idée globale de l'application, grâce au squelette de l'application et aux besoins exprimés en tout début de conception par le client.

[1] http://fpontico.free.fr/ (dernier accès : 25 sept. 08)

4.2. PATRONS D'INTERFACES POUR L'E-PROCURATION

Le catalogue de patrons d'interface pour l'e-Procuration est un support incontournable de la méthodologie de conception que nous proposons. L'utilisabilité d'un tel support passe par la pertinence et l'applicabilité des conseils de conception, ainsi que par leur lisibilité. Pour favoriser la clarté du catalogue, une structure fixe de description des patrons a été choisie, inspirée des patrons de conception classique et adaptée au contexte du processus de conception basé sur *wireframes*. La navigation au sein du catalogue est un autre facteur d'utilisabilité : l'utilisateur du catalogue doit être encouragé à naviguer en son sein, à la recherche de solutions de conception qui correspondent bien à son cas concret. Le catalogue est accessible en ligne à l'adresse **http://fpontico.free.fr** (Pontico 2008), site Web de type Wiki conçu avec l'outil MediaWiki[1]. La Figure 28 montre la page d'accueil de ce catalogue. La consultation du catalogue est en libre accès ; en revanche, les modifications sont de la responsabilité de l'administrateur. Toutefois, tout visiteur peut déposer une suggestion d'ajout ou de modification de patron par email : un formulaire l'y invite, décrivant les informations à donner[2]. Un moteur de recherche sur le catalogue est accessible dans la colonne de gauche.



Figure 28. Page d'accueil du catalogue de patrons (capture d'écran réalisée le 4 juin 2008)

Le catalogue présenté ici n'a pas pour but de présenter de façon exhaustive les patrons d'interface pour la conception d'applications d'e-Procuration. De même, les patrons ont été conçus à partir d'observations sur le terrain mais n'ont en aucun cas été confirmés par un expert en facteurs humains. L'objectif de ce travail est de prouver la faisabilité du principe du catalogue en tant que structure de gestion et de consultation de recommandations ergonomiques, tel qu'il est organisé et opérationnalisé ici. La méthode suivie pour la rédaction des patrons a été la suivante :

- Attribution d'un patron à tout fragment d'interface repéré comme récurrent lors de l'analyse de l'existant (cf. §3.3), inclusion des bons et mauvais exemples relevés lors de cette même analyse ;
- Inclusion d'exemples issus du domaine de l'e-Gouvernement local, régional ou national, mis en place dans des pays différents ;

[1] http://www.mediawiki.org/ (dernier accès : 25 sept. 08)
[2] http://fpontico.free.fr/limesurvey/index.php?sid=72227&lang=fr (dernier accès : 25 sept. 08)

- Complétion des observations faites de ces analyses par des éléments issus de domaines proches (e-Commerce, e-Assurance) lorsque le patron s'y prête (p.ex. boîte d'avancement) :
  - Patrons d'interfaces (Tidwell 2005) (Martijn Van Welie 2007) ;
  - Exemples (bons et mauvais).

4.2.1. Structure systématique d'un patron

Tous les patrons sont décrits de façon analogue dans le catalogue, cette systématisation visant à faciliter l'utilisabilité du catalogue pour l'utilisateur, quelque soit son niveau d'expertise. Des études de psychologie expérimentale ont montré que la mise en place d'indices de structuration sur un texte affecte non seulement la mémorisation de celui-ci mais aussi la prise d'informations (Meyer et Rice 1992). En Interaction Homme Machine également, des recommandations portent sur l'homogénéité des interfaces pour une meilleure utilisabilité (J.M. Christian Bastien et Scapin 1993). La systématisation de la présentation des patrons est ainsi un atout pour l'utilisabilité du catalogue pour les utilisateurs novices auxquelles une structure de mémorisation des informations diffusées est ainsi suggérée. Les utilisateurs expérimentés du catalogue apprécieront quant à eux que cette structuration rende plus efficace leur lecture rapide et la prise d'informations.

Les attributs de description d'un patron suivent une forme classique de patrons de conception pour ce qui concerne les cinq premiers. Ces attributs décrivent en effet les recommandations de mise en œuvre du patron, et s'attachent à couvrir les questions que l'utilisateur pourrait se poser au moment d'appliquer le patron. Le sixième et dernier attribut est plus spécifique à la méthodologie que nous proposons, dans la mesure où il permet au patron d'être un point d'entrée concret dans l'étape de prototypage fonctionnel du cycle de conception.

Description

Les caractéristiques du patron sont données ici, sous forme d'une phrase ou deux, brièvement, en replaçant le patron dans son contexte par exemple.

Description [modifier]

Plusieurs étapes sont à effectuer pour réaliser une procédure. Un wizard guide l'utilisateur à travers ces étapes.

Figure 29. Attribut Description du patron « Wizard »[1]

Exemples

De bons et mauvais exemples d'implémentation du patron sont donnés ici, éventuellement accompagnés de justification de leur choix (notamment lorsque des erreurs de conception classiques y sont relevées). De préférence, les exemples sont tirés des productions de l'entreprise qui utilise le catalogue ou, au moins, du domaine d'application.

[1] http://fpontico.free.fr/index.php?title=Wizard#Description (dernier accès : 25 sept. 08)


Exemples

Je signale mon changement d'adresse [1]
e-Gouvernement national, France
Capturé le 29 juin 2007

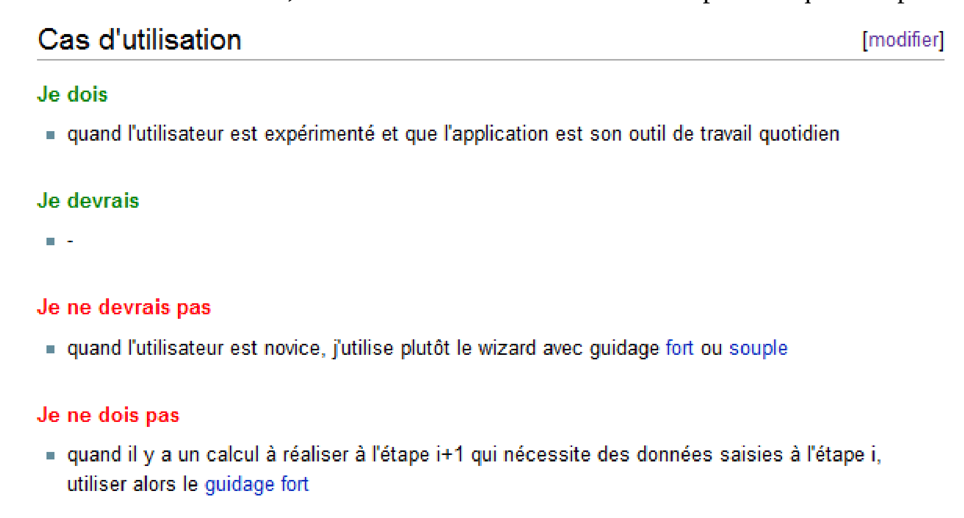
Inscription | Personnes | Organismes | Validation

Les 3 états d'avancement possibles pour une étape sont distingués :
- réalisé avec succès (ici "Inscription")
- en cours (ici "Personnes")
- à réaliser (ici "Organismes" et "Validation")

Figure 30. Attribut Exemple du patron « Boîte d'avancement »[1]

Cas d'utilisation

Les circonstances d'application du patron sont décrites ici. Quand est-il obligatoire ? Quand est-il valable mais pas nécessaire ? Quand est-il déconseillé ? Quand est-il interdit ? Dans les cas de non utilisation, une référence est donnée vers des patrons qui sont préférables.


Cas d'utilisation [modifier]

Je dois

- quand l'utilisateur est expérimenté et que l'application est son outil de travail quotidien

Je devrais

- -

Je ne devrais pas

- quand l'utilisateur est novice, j'utilise plutôt le wizard avec guidage fort ou souple

Je ne dois pas

- quand il y a un calcul à réaliser à l'étape i+1 qui nécessite des données saisies à l'étape i, utiliser alors le guidage fort

Figure 31. Attribut Cas d'utilisation du patron « Wizard avec récapitulatif éditable »[2]

Mise en page

Des recommandations concrètes sont données ici quant à la mise en page, aux informations à faire figurer, au graphisme, aux mises en garde sur des points techniques. Les recommandations viennent en complément du prototype, pour corroborer ce qui y figure et insister sur les éléments qui sont importants.

Mise en page [modifier]

- Quand les utilisateurs sont des travailleurs sociaux, attention au **vocabulaire** manipulé. Parlent-ils de "dossier", de "casier", d'"affaire", de "fiche" ? Respecter ce vocabulaire dans la formulation des items de tableau de bord.
- Quand les utilisateurs sont des particuliers, attention à la **surcharge d'informations**. Privilégiez les objectifs pour lesquels il est le plus probable que l'utilisateur visite l'application (ex. Sur le tableau de bord de gestion des impôts, affichez en premier la déclaration de revenus ou le paiement de la redevance audiovisuelle).

[1] http://fpontico.free.fr/index.php?title=BoiteAvancement#Exemples (dernier accès : 25 sept. 08)
[2] http://fpontico.free.fr/index.php?title=WizardRecapitulatifEditable#Cas_d.27utilisation (dernier accès : 25 sept. 08)

Figure 32. Attribut Mise en page du patron « Tableau de bord »[1]

Ressources

Figurent ici les ressources (article académique, forum technique, livre, autres catalogues de patrons d'interface) qui ont aidé à la définition du patron.



Ressources [modifier]

- Jenifer Tidwell, "Wizard" [4]
- Douglas K. Van Duyne, The Design of Sites, Addison-Wesley 2003 (Order Summary, F7 pattern, page 395)
- Martijn Van Welie, "Shopping cart" [5]

Ce patron propose des recommandations pour l'affichage des achats réalisés sur un site d'e-Commerce. Certaines de ces recommandations peuvent aider à la conception d'un récapitulatif d'application d'e-Gouvernement.

- Martijn Van Welie, "Wizard" [6]

Figure 33. Attribut Ressources du patron « Récapitulatif »[2]

Prototype

Lorsque le patron porte sur un **enchaînement d'écrans**, le prototype est un modèle générique StateWebCharts (Winckler et Palanque 2003) de la navigation telle qu'elle est recommandée sur cet enchaînement d'écrans. Les pages du modèle font référence à des patrons de page existant dans le catalogue.

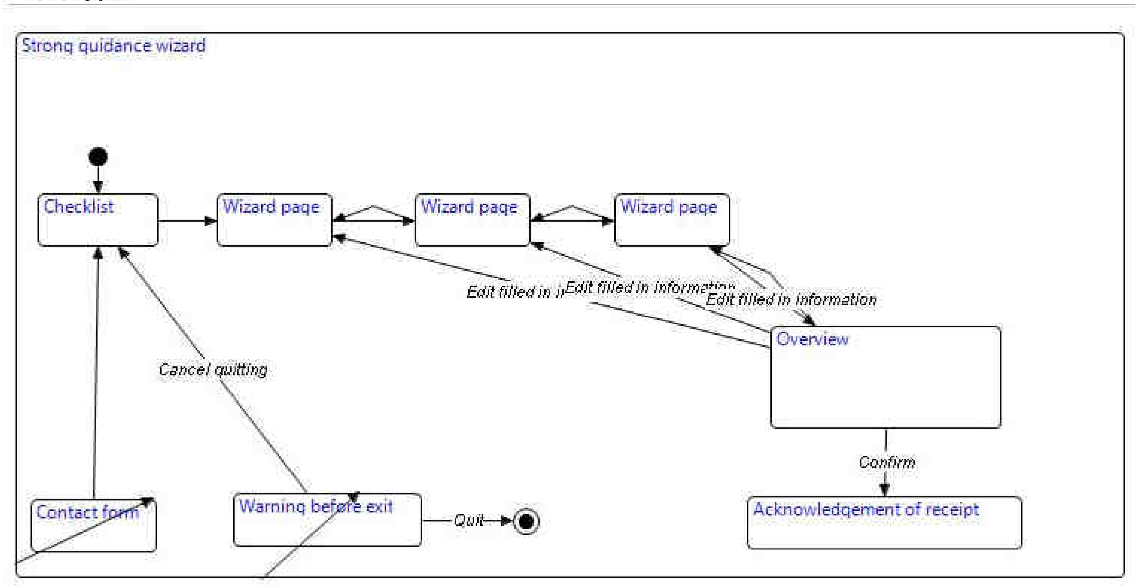


Figure 34. Attribut Prototype du patron « Wizard avec guidage fort »[3]

Lorsque le patron porte sur **une page ou un composant de base**, le prototype est un *wireframe* présentant de façon générique la manière dont le patron doit être implémenté. L'analyste qui utilise le catalogue constitue son propre *wireframe* en combinant plusieurs patrons, et en ajoutant les informations relatives au projet courant. Ces *wireframes* peuvent être qualifiés de prototypes de fidélité moyenne, considérant les critères de McCurdy (McCurdy et coll. 2006) : affinement visuel (bon), largeur de la couverture des fonctionnalités (bonne), profondeur de la

[1] http://fpontico.free.fr/index.php?title=TableauDeBord#Mise_en_page (dernier accès : 25 sept. 08)
[2] http://fpontico.free.fr/index.php?title=Recapitulatif#Ressources (dernier accès : 25 sept. 08)
[3] http://fpontico.free.fr/index.php?title=WizardGuidageFort#Prototype (dernier accès : 25 sept. 08)

prise en compte des fonctionnalités (faible), richesse de l’interactivité (moyenne), richesse du modèle de données (faible).

Figure 35. Attribut Prototype du patron "Page de wizard"

4.2.2. Organisation du catalogue de patrons

L’étude réalisée au chapitre précédent (cf. §3.2), consécutivement au relevé de fragments d’interface récurrents, a permis de dégager les atouts d’une organisation de patrons efficace. La difficulté de la mise à disposition de patrons est en effet de rendre accessible et de faciliter la navigation au sein de cette base de connaissances en matière de conception. La conclusion stipulait que l’organisation des patrons devait fournir à la fois une structure hiérarchique et des possibilités de navigation alternative (références croisées entre éléments d’interface, groupement d’éléments proches, comparaison de points de vue).

Il fallait également garantir qu’un passage à l’échelle n’entame en rien l’efficacité de l’organisation, un catalogue de patrons étant fortement évolutif par nature. La structure hiérarchique de notre catalogue est détaillée dans un premier temps, puis nous présentons une proposition d’ontologie qui satisfasse les besoins en termes de navigation, tout en conservant ses atouts lors de l’évolution du volume de patrons dans le catalogue.

La manière la plus élémentaire de classer les patrons d’interface est de les organiser suivant l’ordre de lecture qui surviendrait dans un processus de conception linéaire. Les questions de conception s’articulent suivant un degré de granularité de plus en plus fin : les questions concernant la globalité de l’application sont d’abord étudiées, suivies de questions sur des éléments d’interface de plus en plus pointus pour en arriver aux *widgets* de l’interface. Le mode de classement retenu est ainsi le même que celui qui régissait les fragments d’interface récurrents dans la section 3.3.

Les **patrons d’enchaînement d’écrans** sont regroupés et apparaissent en premier, traitant de problèmes d’articulation des pages au sein de l’application. La structure hypertexte est considérée comme étant la vue la plus globale sur l’application. Les **patrons de page** arrivent ensuite, figurant les pages apparaissant par ailleurs dans les patrons d’enchaînements d’écrans. La méthodologie a pour objectif de soutenir les débuts de la conception, et en particulier la phase d’analyse fonctionnelle. Les choix de conception d’interface ne vont donc pas jusqu’au niveau des *widgets* et le degré de détail maximum sera celui des **patrons de composants de base**. Ces composants de base ont un rôle dans l’interaction avec l’utilisateur, dans la saisie ou la présentation d’informations, l’offre de navigation ou encore le guidage. Leur niveau de détail est celui de l’analyse fonctionnelle : il s’agit de définir des menus ou des formulaires, et, de manière additionnelle, leur comportement ou les contraintes les régissant. Les patrons de chaque catégorie comportent les mêmes attributs de description, à la différence que le prototype d’un enchaî-

nement d'écrans sera de nature différente de ceux d'une page ou d'un composant de base (cf. §4.2.1).

4.3. ÉVOLUTION DU CATALOGUE

Le catalogue de patrons d'interfaces est un objet fortement évolutif, dont les utilisateurs sont les principaux contributeurs. Lorsque l'analyste rencontre un problème de conception d'interface qui ne trouve pas de réponse dans le catalogue, il fournit un effort de recherche considérable pour, en dépit de cette absence, produire un modèle de navigation ou un *wireframe* de page ou de composant de base qui satisfasse le problème rencontré. Cet effort mérite d'être capitalisé et partagé avec la communauté d'utilisateurs du catalogue. L'analyste peut alors suggérer l'ajout d'un patron au catalogue.

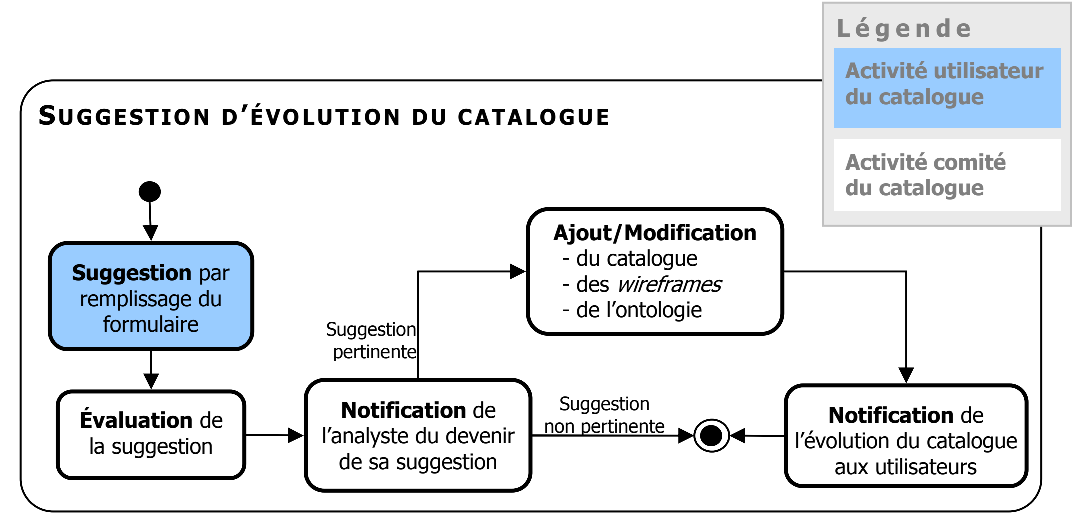


Figure 36. Procédure de mise à jour du catalogue de patrons

4.3.1. Suggestion par remplissage du questionnaire

Les suggestions sont traitées de la même façon pour l'ajout ou la modification d'un patron du catalogue. Idéalement, une suggestion est formulée suite au succès du déploiement d'un fragment d'interface nouvellement produit : l'équipe de conception a reçu de bons retours sur l'application et l'application a été déployée et est utilisée depuis un temps qui l'a menée à la stabilité dans la forme et le contenu. C'est alors que l'analyste suggère l'ajout d'un fragment d'interface qui a contribué au succès de l'application et ne figure pas dans le catalogue. Toutefois, il est acceptable de recevoir des suggestions dès la fin de la spécification : quoique le fragment n'ait pas encore été déployé, les raisonnements qui ont conduit à sa mise en place sont encore clairs dans l'esprit de son concepteur. Ces raisonnements sont précieux pour l'ajout au catalogue.

À partir du formulaire référencé depuis le catalogue[1] (cf. Figure 37), et réalisé grâce à l'outil LimeSurvey[2], l'utilisateur du catalogue transmet ses suggestions au comité responsable du catalogue qui statuera sur cette proposition. Il est invité, à l'issue de son formulaire, à envoyer par courrier électronique un prototype du patron suggéré. Son email est requis pour le tenir au courant du devenir de sa suggestion. Les informations demandées dans le formulaire sont les suivantes :

- **Nature de la suggestion** : ajout ou modification ? (champ obligatoire) :
  en cas de modification, la suite des questions est remplacée par : 1) la saisie de

[1] http://fpontico.free.fr/limesurvey/index.php?sid=72227&lang=fr (dernier accès : 25 sept. 08)
[2] http://www.limesurvey.org/ (dernier accès : 25 sept. 08)

l'URL du patron à modifier ; 2) un champ de commentaire libre ; 3) la saisie de l'email du contributeur.

- **Titre du patron** (champ obligatoire) tel qu'il apparaîtrait dans le catalogue ;
- Brève description du patron (champ obligatoire) ;
- **Exemples bons ou mauvais** (champ facultatif) : il est demandé de préférence de fournir des exemples issus d'applications d'e-Procuration, et de signaler l'URL de l'exemple ;
- **Cas d'utilisation** (champ facultatif) : situations dans lesquelles le patron doit / devrait / ne devrait pas / ne doit pas être appliqué ;
- Recommandations de mise en page (champ facultatif) ;
- **Ressources complémentaires** (champ facultatif) : documents ayant aidé le contributeur à mettre en place le patron en situation réelle, ou à compléter le formulaire de suggestion ;
- **Adresse email** (champ obligatoire).

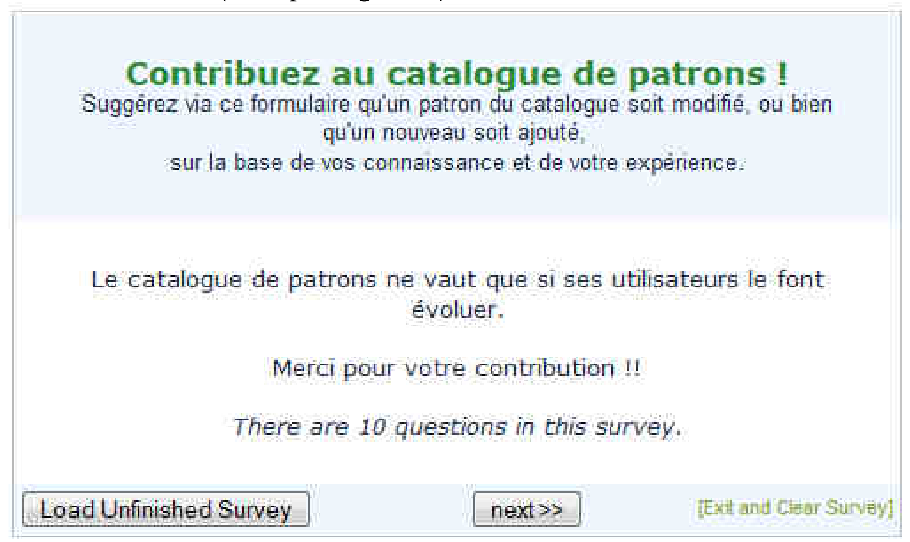


Figure 37. Formulaire de suggestion d'évolutions du catalogue

Volontairement, la plupart des champs du formulaire sont facultatifs, pour permettre au contributeur de suggérer l'ajout ou la modification d'un patron de façon informelle et rapide. Il est nécessaire de suggérer à la personne qui remplit ce formulaire que ses réponses ne seront pas jugées, mais considérées avec intérêt. Des exemples de réponse figurent systématiquement.

4.3.2. Évaluation et prise en compte de la suggestion

Le comité est composé principalement d'experts en utilisabilité et d'analystes confirmés, aptes à trancher sur des questions de conception. Il doit en effet pouvoir formaliser la suggestion de l'analyste pour l'intégrer au catalogue. Quelle que soit la pertinence de la proposition, une réponse est donnée à l'analyste qui la suggère, afin de le conforter dans son élan de contributeur.

Si la proposition est prise en compte, un avertissement est communiqué à la communauté des utilisateurs du catalogue, pour les inciter à consulter la version optimale du catalogue, et insister sur l'aspect évolutif de celui-ci, et leur responsabilité dans cette évolution. La base de *wireframes* intégrée à l'outil eGovIPM doit être mise à jour également : le catalogue et l'outil doivent impérativement évoluer simultanément. L'ontologie est mise à jour de même, par le comité responsable du catalogue.

4.3.3. Notifications

Le comité de gestion du catalogue envoie un courrier électronique au contributeur 1) à réception de la suggestion, 2) dès qu'elle est portée à l'étude, 3) dès qu'elle est rejetée ou acceptée, 4) quand elle est éventuellement publiée. La procédure est la même si l'analyste souhaite suggérer l'ajout d'un nouveau patron ou la modification d'un patron existant.

En cas d'ajout ou de modification effective d'un patron, la rubrique « Actualités » de la page d'accueil du catalogue est mise à jour (cf. Figure 38). Les modifications sont classées par ordre anti-chronologique ; les ajouts et les modifications majeures sont mis en exergue. Tous les mois, un email est envoyé à l'ensemble des utilisateurs connus du catalogue pour leur résumer les mises à jour effectuées.

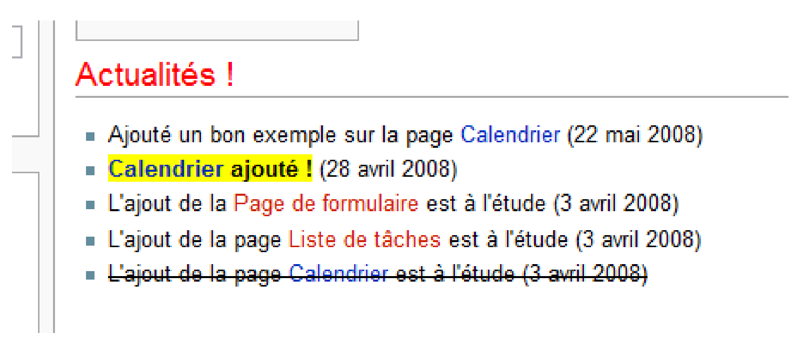


Figure 38. Section actualités de la page d'accueil du catalogue de patrons (capture d'écran réalisée le 2 juin 2008)

4.4. SCÉNARIOS D'UTILISATION

La méthode de conception pour l'e-Gouvernement que nous proposons est illustrée dans cette section, sur une étude de cas fictive. La mairie de Goville souhaite concevoir le site Web de la cantine de l'école communale. Pour ce faire, elle fait appel à une entreprise de technologies de l'information qui utilise notre méthodologie. Les besoins exprimés sont succincts, et c'est sur leur base que l'analyste doit produire une première proposition d'interface :

- Accès sécurisé pour les parents qui souhaitent consulter les repas auxquels sont inscrits leurs enfants, ajouter ou supprimer des repas ;
- L'identifiant et le mot de passe leur sont remis en mains propres, à l'école ;
- On peut ajouter ou supprimer un repas jusque trois jours avant la date du repas ;

À partir de cette étude de cas, cette section montre comment notre méthode satisfait les trois critères évoqués à la fin de notre état de l'art des méthodes de conception d'interfaces centrées utilisateur :

- Soutenir la conception :
  - Support à la conception de *wireframes* de pages, à partir du catalogue de patrons ;
  - Support à la définition de la navigation par un modèle StateWebCharts, à partir du catalogue de patrons et de l'outil eGovIPM (cf. chapitre 6) ;
- Favoriser la communication : génération d'un squelette de l'application qui puisse être discuté avec le client, sur la base des *wireframes* de page et de l'enchaînement d'écrans, et grâce à l'outil eGovIPM ;
- Capitaliser la connaissance : lorsque l'analyse révèle des éléments n'apparaissant pas dans le catalogue, le concepteur suggère un ajout de patron.

4.4.1. Soutenir la conception

Sur la base des besoins exprimés par le demandeur, l'analyste réalise les artefacts de conception décrits dans cette section. Pour ce faire, il utilise eGovIPM (le détail de cette réalisation sera fourni dans le chapitre 6 consacré à l'outil), les patrons figurant dans le catalogue de patrons et un outil éditeur de dessin annexe, de type Visio. L'étude de cas n'est que partiellement traitée et nous ne présentons ici que la page concernant la gestion des repas des enfants.

Définition d'un **wireframe** *de page*

Pour concevoir son *wireframe* de page, l'utilisateur navigue sur le catalogue de patrons. Aucun patron de page ne correspond à son cas précis : définir une page permettant la gestion des repas d'un ou plusieurs enfants. Mais cette page a un axe central très fort : la manière dont les repas doivent être présentés. L'analyste cherche d'abord un patron traitant de « dates » : il saisit ce mot-clef dans le moteur de recherche interne du catalogue. La Figure 39 montre le résultat de cette recherche, qui retourne notamment le patron « Calendrier » qui contient le mot « date ».



Figure 39. Résultat de la recherche du mot clé « date » sur le catalogue de patrons (capture d'écran réalisée le 22 mai 2008)

C'est le patron de composant de base « **Calendrier**[1] » qui sert de base à la conception de cette page. Comme suggéré par le prototype de ce patron (cf. Figure 40), un enfant inscrit à un repas est représenté par une couleur. Les dates du calendrier ne sont pas cliquables, compte tenu de la complexité des données à gérer : sur une seule date, on peut en effet ajouter ou supprimer un repas pour chacun des enfants. Il est donc proposé à l'utilisateur de cliquer sur un des liens « Modifier » pour modifier les inscriptions aux repas d'un ou plusieurs de ses enfants. Cette pop up (non présentée ici) propose l'ajout ou la suppression de dates, les dates étant grisées, rendues inactives jusque trois jours ouvrables après la date du jour. En parallèle, l'analyste consulte les patrons d'enchaînements d'écrans : pour ce cas d'étude dont la procédure est simple, il s'agira

[1] http://fpontico.free.fr/index.php?title=Calendrier (dernier accès : 25 sept. 08)

d'un « **Tableau de bord** » avec le calendrier général de la Figure 40 comme nœud central, depuis lequel seront accessibles une tâche « Modifier » par enfant, plus une pour l'ensemble des enfants.

Une fois le cœur de la page élaboré, l'analyste l'enrobe avec un cadre générique pour la page, inspiré par le patron de composant de base « **En tête et Pied de page**[1] », dont le prototype est reproduit sur la Figure 41.



Figure 40. Prototype du patron de composant de base « Calendrier »

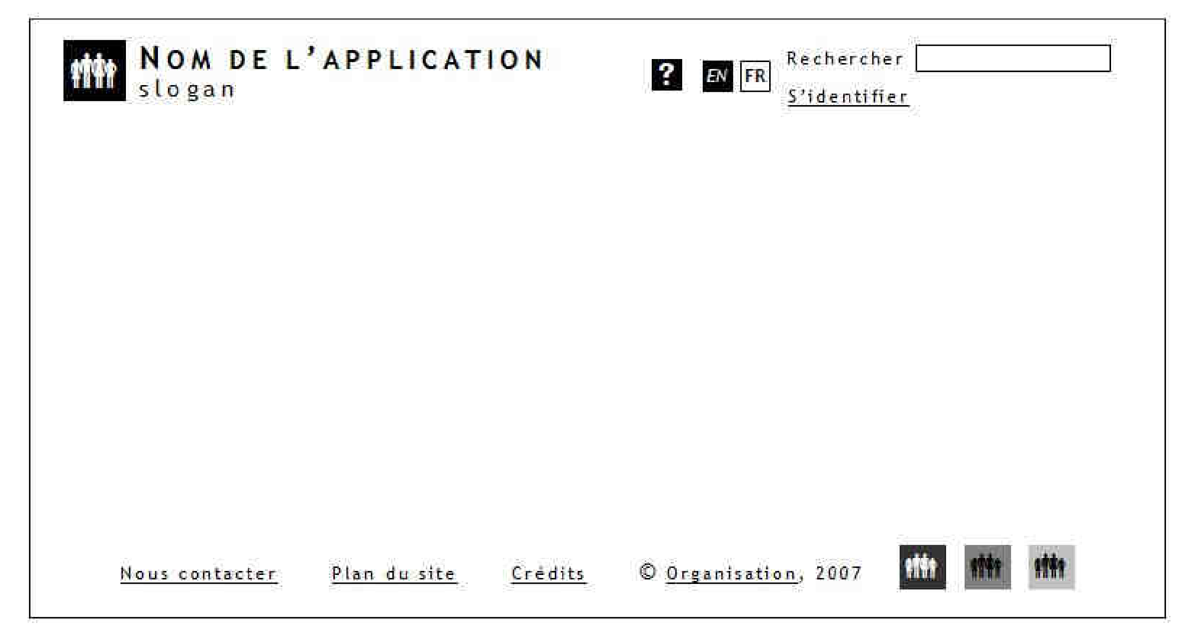


Figure 41. Prototype du patron de composant de base « En-tête et pied de page »

[1] http://fpontico.free.fr/index.php?title=EnTeteEtPiedPage (dernier accès : 25 sept. 08)

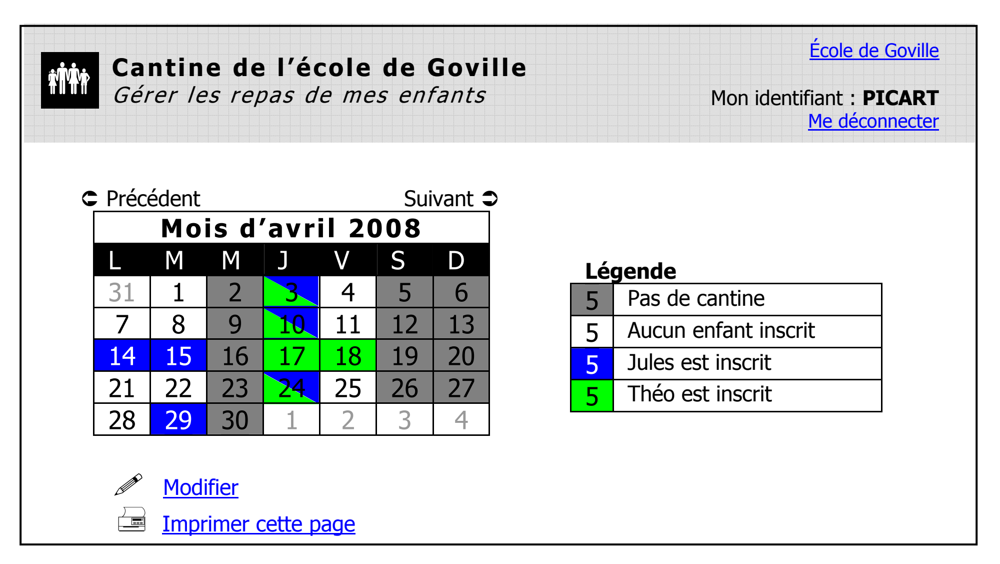


Figure 42. *Wireframe* de la page « Repas des enfants »

Définition de la navigation

Nous reviendrons en détail sur les étapes de réalisation du modèle de navigation de la Figure 43 dans le chapitre 6 dédié à l'outil eGovIPM. Le patron de base est le canevas « **Tableau de bord** » d'eGovIPM, comme il avait été repéré lors de la conception des *wireframes* de pages. Suite au login de l'utilisateur, le parent d'élève dispose d'un tableau de bord. Cette page (cf. Figure 42) comprend un calendrier des repas auxquels les enfants sont inscrits et des liens permettant de modifier les repas de chaque enfant, ou de tous les enfants en une seule fois. Depuis toute page de l'application, un lien est fourni vers le site de l'école de Goville.

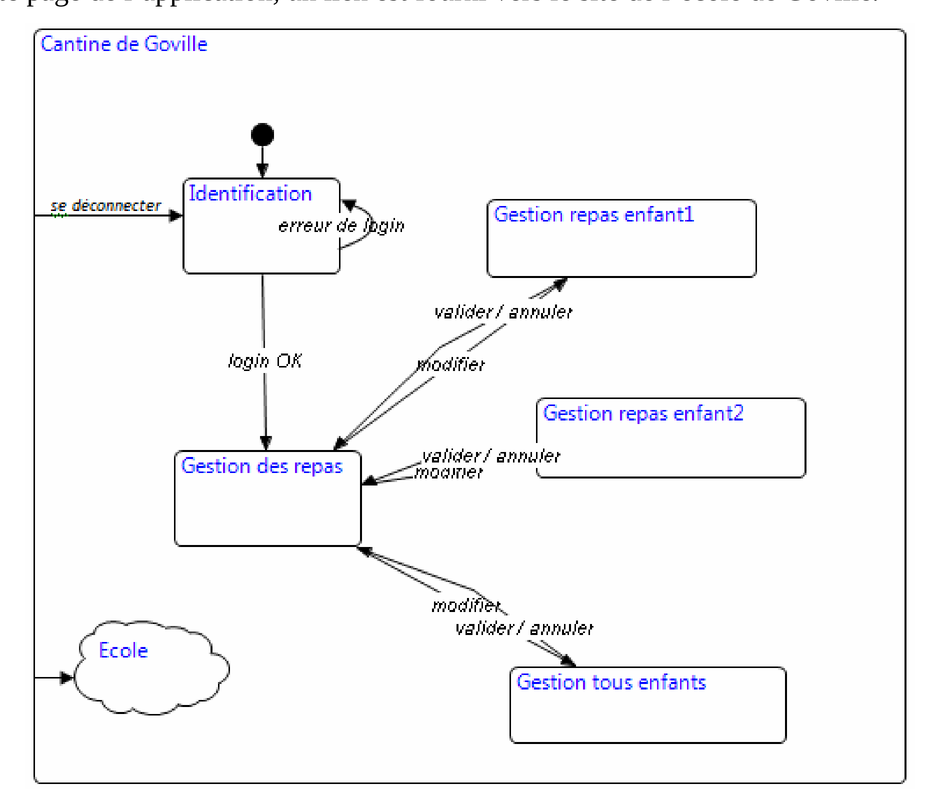


Figure 43. Modèle de navigation SWC du site Web de la cantine de Goville

4.4.2. Favoriser la communication

Pour sa rencontre avec le client, l'analyste a généré avec eGovIPM (cf. chapitre 6) les pages XHTML de l'application à partir du modèle de navigation SWC de la Figure 43 et des images de ses *wireframes* de page personnalisés. Il parcourt le modèle de navigation par le biais d'un navigateur Web, profitant de chaque nouvelle page pour en expliquer les fonctionnalités.

La Figure 44 ci-après montre la page d'identification générée automatiquement depuis le modèle StateWebCharts. L'analyste n'a pas jugé bon, dans un premier temps, de personnaliser le *wireframe* de cette page : c'est donc le *wireframe* générique du catalogue de patrons[1] qui apparaît. La Figure 45 a été générée automatiquement également, mais l'analyste a spécifié sur le modèle StateWebCharts de la navigation qu'un *wireframe* personnalisé était disponible : il apparaît alors.

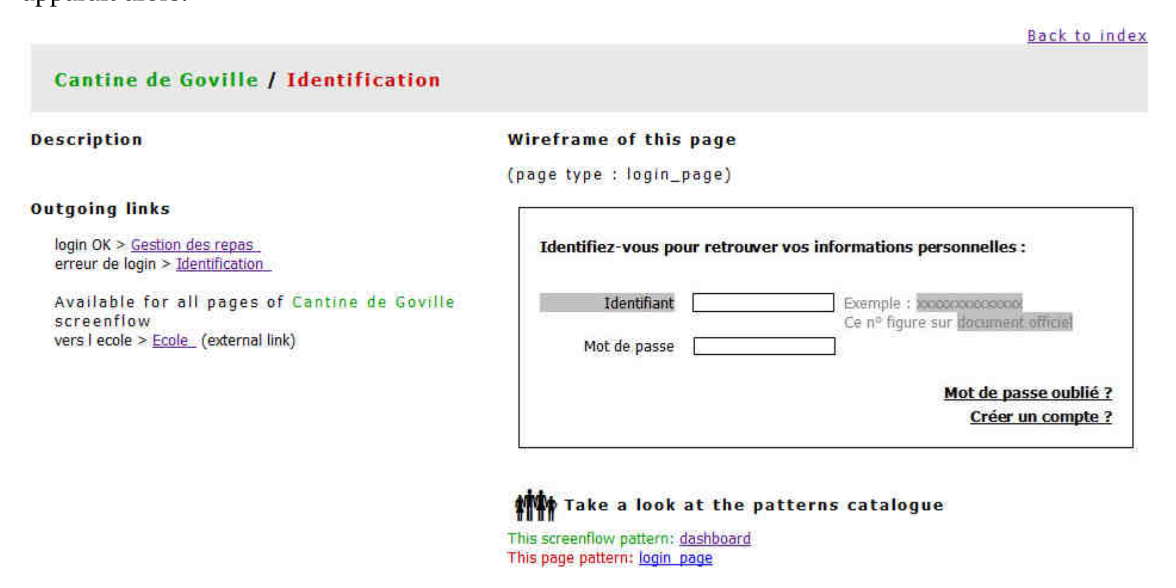


Figure 44. Page XHTML du squelette de l'application généré par eGovIPM : Identification

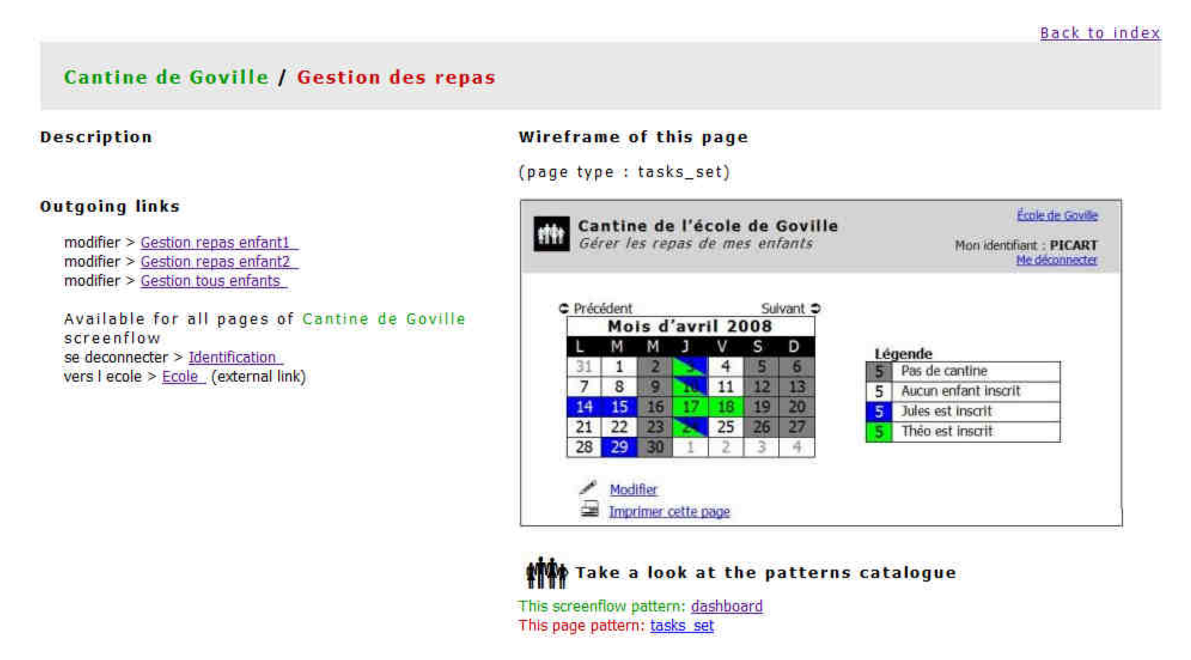


Figure 45. Page XHTML du squelette de l'application généré par eGovIPM : Gestion des repas

[1] http://fpontico.free.fr/index.php?title=Login (dernier accès : 25 sept. 08)

Le client est satisfait des modes d'interaction choisis et des informations dispensées. Cependant, à la vue de cette proposition, une idée émerge : celle d'ajouter une fonctionnalité permettant à un parent de payer la cantine pour le trimestre passé, par carte de crédit. Il demande à l'analyste si un tel ajout est possible en peu de temps, afin de ne pas prendre trop de retard dans le projet. L'analyste accepte et entame la deuxième définition de l'interface, pour y ajouter ce module de paiement par carte de crédit.

4.4.3. Capitaliser la connaissance

Le client du site Web de la cantine a demandé l'ajout d'un module de paiement par carte bleue sur le site. L'analyste a fourni un effort considérable pour produire un *wireframe* de page satisfaisant pour cette page, qui ne figurait pas dans le catalogue. Afin que cet effort soit formalisé par le comité de gestion du catalogue de patron, et éventuellement intégré à celui-ci, il renseigne les informations suivantes par le biais du questionnaire. Puis, par manque de temps, il ne rend pas générique le *wireframe* tel qu'il l'a conçu pour le projet de la cantine (cf. Figure 47), et l'envoie tel quel, par courrier électronique à l'adresse indiquée sur le catalogue.

Titre du patron

Paiement par carte de crédit.

Description

Un particulier souhaite régler un montant via l'application, au moyen de sa carte de crédit.

Exemples

Site d'e-Commerce Etam http://www.etam.fr (dernier accès : 25 sept. 08)



Figure 46. Bon exemple pour la suggestion de patron « Paiement par carte de crédit »

Cas d'utilisation

Il **faut** utiliser ce patron dès qu'un particulier souhaite payer par carte de crédit.

Il **ne faut pas** utiliser ce patron lorsqu'une entreprise doit payer ; préférer alors le remplissage de bons de commandes.

Mise en page

Utiliser les logos officiels des différentes cartes de crédit.

Afficher clairement le logo de la banque de l'application.

Rappeler en tête de page le montant à payer.

Autoriser l'annulation du paiement avant qu'il n'ait été validé.

Ressources

–

Prototype



Figure 47. Wireframe de la page « Paiement du trimestre passé »

4.5. Discussion et perspectives

Nous avons proposé dans ce chapitre une méthode de conception basée sur des patrons d'interfaces pour l'e-Gouvernement. Le support permettant le recueil des patrons a été formalisé par le biais d'un catalogue hypertexte, et une procédure de mise à jour de ce support est suggérée. L'évolutivité d'un catalogue de patrons étant l'un des signes de sa vitalité et de son utilité, les utilisateurs doivent en effet être encouragés à soumettre des ajouts de patrons. Notre méthode entre dans le cadre d'un processus de conception basé sur des prototypes moyenne fidélité, incluant des prototypes moyenne fidélité sous forme de modèles de navigation et de *wireframes*. Des pistes d'amélioration de cette méthode restent à explorer, en particulier au niveau de la navigation au sein du catalogue, vers une offre de recherche de solution optimale pour le concepteur.

4.5.1. Ontologie des concepts des patrons

Le classement des patrons par granularité (cf. §4.2.2) est une manière simpliste de présenter aux concepteurs le type de problèmes de conception auxquels ils se réfèrent. La linéarité dans la conception n'est en effet ni réaliste ni souhaitable : les mécanismes itératifs permettent un affinement des propositions et autorisent une certaine souplesse dans la conception. Une voie de navigation alternative à la voie suivant la granularité des patrons doit donc être fournie aux

utilisateurs du catalogue. L'état de l'art de l'organisation des patrons de conception (cf. §3.2) avait mis en avant le fait qu'une ontologie tisserait entre patrons des liens sémantiques forts. Cela permettrait par un mécanisme de navigation unique, de proposer des liens entre patrons concernant les mêmes éléments d'interface, ou bien apparaissant dans des circonstances similaires, ou encore destinés au même public. La construction de l'ontologie requiert une standardisation des termes employés pour la décrire, afin que des liens puissent être tissés par similarité.

L'ontologie contiendrait des éléments qui ne figurent pas forcément dans le catalogue. Cela permettrait d'éviter la surcharge d'informations sur le catalogue, et de rendre les patrons du catalogue plus lisibles. Cependant, l'information est stockée dans l'ontologie, et permet de tisser des liens entre patrons aux informations similaires, ou d'effectuer sur ces informations des recherches avancées pour un utilisateur expert. Parmi ces informations masquées sur le catalogue mais présentes dans l'ontologie préliminaire que nous avons établie, citons :

- **Sens d'interaction** : le fragment d'interface décrit par le patron sert-il à la saisie d'informations de la part de l'utilisateur ou bien à la diffusion d'informations ? ;
- **Fréquence d'utilisation** : le fragment d'interface décrit par le patron est-il recommandé pour une fréquence particulière d'usage ? Par exemple, un patron offrant un fort guidage est adapté à une utilisation très ponctuelle, mais n'est pas recommandé pour une utilisation quotidienne ;
- **Inclusion à d'autres patrons** : pour un patron de granularité donnée, liste des patrons de granularité supérieure auxquels il peut être inclus ;

La Figure 48 présente une proposition d'attributs définissant une page, ces attributs étant des chaînes de caractères ou des références à des instances d'autres classes (personne, cas d'utilisation, patron, etc.) L'ontologie a été construite avec l'outil Protégé (Stanford Center for Biomedical Informatics Research 2008).

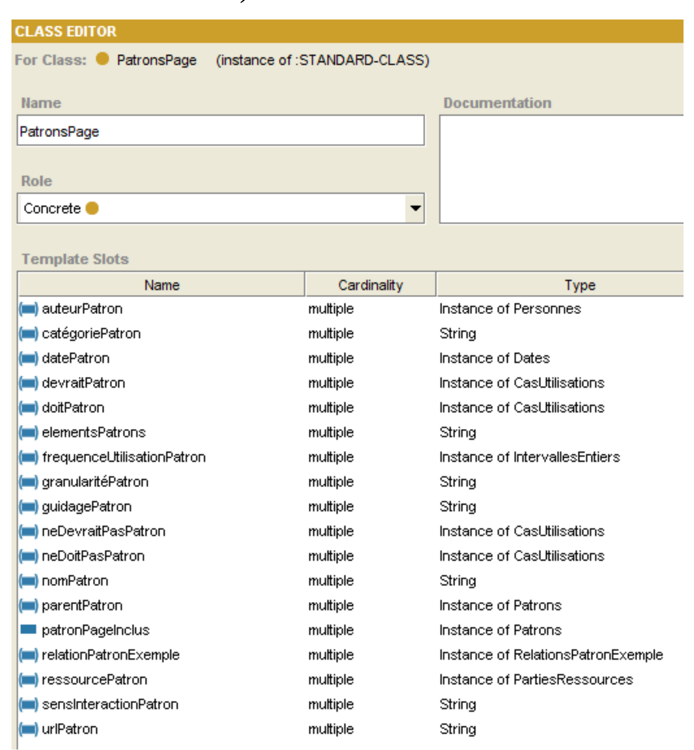


Figure 48. Exemple de définition d'attributs standardisés pour une page dans l'ontologie du catalogue (classe « PatronPage » définie dans l'outil Protégé)

Des outils de visualisation d'ontologie sont enfichables dans l'outil Protégé, et parmi eux TGVizTab *TouchGraph Visualization Tab*[1] (Alani 2003) : c'est grâce à cet outil qu'un fragment de l'ontologie est présentée sur la Figure 49. L'affichage est paramétrable et une recherche est possible sur les valeurs d'instances figurant dans l'ontologie. Ici certains nœuds ont été masqués pour plus de lisibilité (p.ex. auteur du patron), et il a été spécifié un radius de valeur 1, c'est-à-dire que seuls les nœuds liés directement (c'est-à-dire par 1 transition) au nœud central (Page de wizard) figurent sur le graphe. Nos projets sont d'étendre le catalogue de patrons avec un tel outil d'aide à la navigation pour ajouter des liens entre patrons autres que leur catégorie d'appartenance, et donner du sens à ces liens. Des études complémentaires et des évaluations avec des utilisateurs seraient requis pour l'intégration d'un tel outil d'aide à la navigation dans un catalogue de patrons, parmi lesquels : choix des informations à présenter, modes d'interaction avec la carte, degré de personnalisation offert, intégration concrète à une application Web.

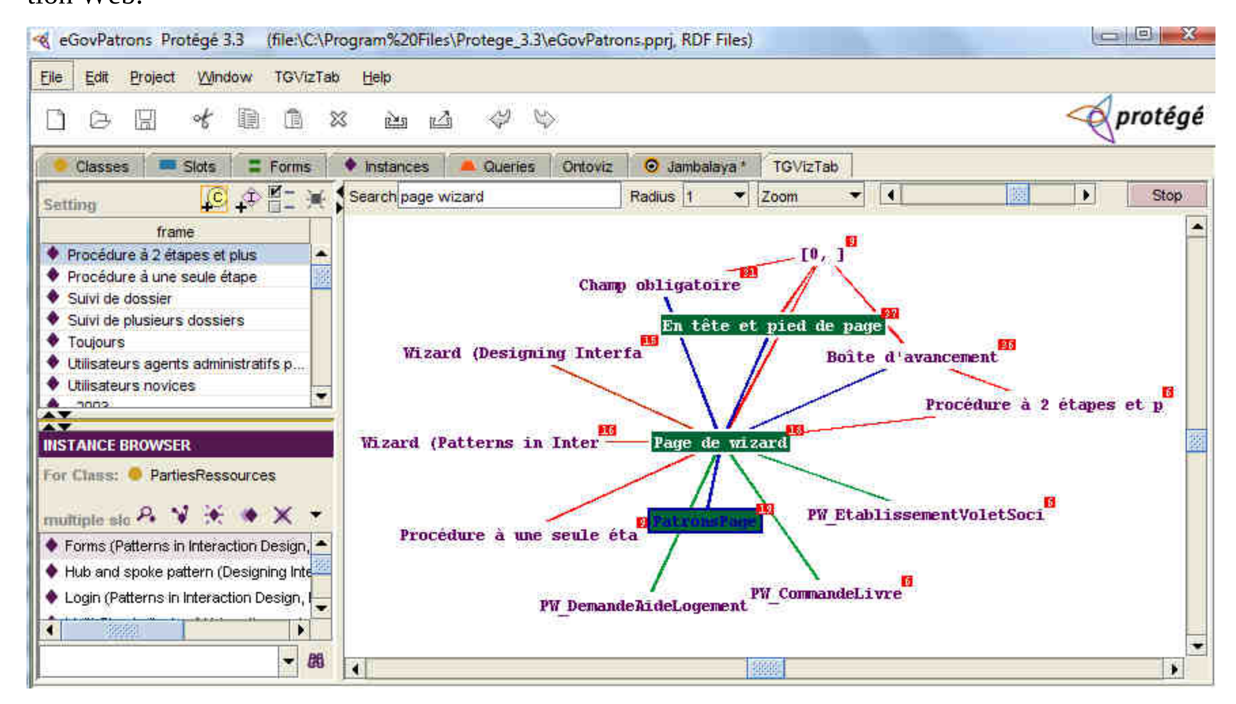


Figure 49. Extrait de la visualisation de l'ontologie des éléments du catalogue, générée par TGVizTab

La visualisation des concepts de l'ontologie doit faire l'objet d'une étude : de nombreux travaux ont été réalisés sur la visualisation des ontologies en recherche d'informations et en Interaction Homme Machine (Katifori et coll. 2007). Les moyens d'intégrer une telle visualisation doivent être étudiés également, y compris en ce qui concerne le dosage des informations à présenter à l'utilisateur. Les concepts de l'ontologie que nous avons présentée sont nombreux et détaillés : le niveau de détail requis pour la conception devra être réfléchi. Des évaluations avec des utilisateurs sur la navigation au sein d'ontologies donneront des pistes d'approfondissement.

4.5.2. *Case-based reasoning*[2]

Le raisonnement basé sur les cas (CBR) est une technique de capitalisation et de réutilisation de connaissances issue de recherches en Intelligence Artificielle (Shiu et Pal 2004). Le principe est, à partir d'une base d'expériences connues et résolues avec succès, de fournir une solution à un problème qui se pose. Les solutions existantes sont analysées par analogie avec le problème posé, de sorte que le système CBR sélectionne et présente au concepteur seulement une partie d'entre elles, les plus pertinentes. Ces théories ont été appliquées au domaine de la conception d'interfaces (Joshi et McMillan 1996), et pourraient être une source d'inspiration

[1] Onglet de visualisation de graphe interactif
[2] Raisonnement basé sur les cas

pour faciliter la recherche dans notre catalogue des patrons adéquats lors de la conception d'une application d'e-Procuration.

Les concepts établis dans la section précédente (cf. §4.5.1) comme devant figurer dans l'ontologie des concepts pourraient servir à la définition formelle des « cas » pour la mise en place d'un tel système. L'évolutivité et la souplesse d'un tel système doivent toutefois être étudiés, les patrons étant amenés, par nature, à occuper un volume croissant. De plus, la navigation au sein du catalogue doit être guidée et personnalisée, mais conserver l'aspect exploratoire : cela permet à l'utilisateur du catalogue de prendre connaissance de son contenu, de se l'approprier. La sélection automatique de solutions doit être un des moyens de naviguer dans le catalogue, mais pas l'unique.

4.5.3. Validation de la méthode

Notre catalogue a été conçu à titre illustratif, pour une exploitation académique d'un cadre de description systématique et d'organisation de patrons d'interfaces pour l'e-Procuration qui soit libre de droits[1]. Notre catalogue n'a pas été déployé en conditions réelles de conception. Nous avons toutefois pu évaluer des hypothèses de recherche à son sujet, sur son acceptabilité en entreprise : nous avons mené une observation des comportements d'usage d'une méthode similaire à la nôtre qui a, elle, été déployée dans un cadre de travail réel, le StyleGuide.

Le StyleGuide est un catalogue de patrons d'interfaces développé par SmalS, dont nous avions contribué à l'élaboration lors d'une observation exploratoire participante en octobre 2006 (cf. chapitre 3). En 2008, parvenu à maturité, le StyleGuide est utilisé par une trentaine de personnes, principalement analystes et développeurs d'interface. Après avoir cerné les ressemblances et dissemblances entre notre méthode et le StyleGuide, une évaluation a été menée sur celui-ci, pour valider les orientations prises par notre méthode, sur son utilisabilité et son acceptabilité en conditions industrielles de travail. Le chapitre 5 expose le déroulement de cette évaluation et ses conclusions.

[1] Notre catalogue est en effet libre de tout *copyright*, à la différence du StyleGuide qui est la propriété intellectuelle de l'entreprise SmalS.

5. Évaluation de la méthode par ses utilisateurs

Résumé

Une méthodologie de conception doit être utilisable, à la manière d'un système interactif. La méthode doit être en accord avec les besoins et les compétences de ses utilisateurs, à savoir les concepteurs. Cette section présente le protocole et les résultats d'une évaluation qui a été menée avec les utilisateurs d'un support à la réutilisation de solutions à des problèmes d'interface récurrents. Les sujets utilisent cet outil sur des projets pilote depuis quelques mois, en situation réelle de conception d'applications d'e-Gouvernement. Ils sont convaincus du bien fondé de la méthodologie accompagnant cet outil, mais rencontrent des difficultés à l'utiliser : ces difficultés sont un frein à une appropriation complète de l'outil. Grâce à des entretiens et des observations de scénarios, des pistes d'amélioration ont pu être dégagées pour une meilleure intégration de la méthode aux activités de travail de ses utilisateurs.

Notre méthode (cf. chapitre 4) vise les mêmes objectifs que le StyleGuide, un catalogue de patrons d'interfaces pour l'e-Procuration développé par SmalS : guidage de la conception vers une application utilisable, standardisation des interfaces, capitalisation et opérationnalisation des connaissances de conception. Une évaluation du StyleGuide auprès de ses utilisateurs a été conduite dans le but d'observer et d'améliorer leurs conditions d'usage. L'observation portait sur des activités de conception provoquées, à réaliser avec le support du StyleGuide. D'autres tâches visaient à éprouver des hypothèses formulées sur les points communs entre notre méthode et celle du StyleGuide. À l'issue de ces évaluations, des suggestions ont pu être faites pour améliorer l'utilisabilité et l'acceptabilité du StyleGuide ; des perspectives d'évolution de notre méthode ont été aussi dégagées.

La section 5.1 met en avant les similarités entre le StyleGuide et la méthode que nous proposons dans cette thèse : support à l'analyse fonctionnelle par le prototypage moyenne fidélité, standardisation des interfaces, capitalisation et mise à disposition de la connaissance en conception d'interfaces. C'est ainsi que seront justifiés les apports, pour la validation et les perspectives concernant notre méthodologie, de l'évaluation de cette méthode. La section 5.2 décrit les modalités de fonctionnement et d'opérationnalisation de cette méthode dans le cadre de l'entreprise SmalS. La section 5.3 présente les hypothèses de recherche qui nous ont permis d'établir l'étude évaluative décrite dans la section 5.4. Les résultats sont rapportés dans la section 5.5. À leur lumière, nous proposons dans la section 5.6 des perspectives d'amélioration du StyleGuide, pour la correction des problèmes d'utilisabilité relevés. Une mise en correspondance des interprétations vis-à-vis de notre propre méthode clôt également ce chapitre.

5.1. Contexte de l'évaluation

Lors de l'observation exploratoire décrite dans le chapitre 3, nous avons participé aux prémices de la mise en place d'une méthodologie de support à la spécification et à la conception d'interface : le StyleGuide. Le StyleGuide est un outil de stockage de solutions à des problèmes de conception d'interface récurrents en e-Procuration. Il est exclusivement, à l'heure actuelle, à usage interne à SmalS ; au sein de l'entreprise, il est ainsi utilisé par des analystes, des graphistes, des éditeurs de contenu et des développeurs d'interface (cf. §5.2 pour plus de détails). Le StyleGuide est ainsi une illustration typique de la volonté de réutiliser les compétences acquises par les employés de l'entreprise, et de standardiser les interfaces pour en assurer l'utilisabilité. Cela est fait par le biais de recommandations concrètes et guidant fermement le concepteur. Leur méthode basée sur le StyleGuide est déployée depuis plus d'un an en projet pilote, et depuis quelques mois de façon plus large : évaluer l'utilisabilité de cette méthode est apparu

comme un moyen d'évaluer le bien-fondé et la faisabilité de la nôtre. Notre philosophie de conception d'interfaces centrée sur l'utilisateur et facilitant la communication est en effet commune à celle du StyleGuide, malgré les différentes voies d'approfondissement suivies depuis octobre 2006 :

- Pour SmalS, opérationnaliser rapidement le catalogue : l'équipe de conception du catalogue souhaitait y faire figurer tous les éléments permettant la conception à partir de modules réutilisables, notamment au niveau technique. Des scripts JavaScript, des feuilles CSS et des pages XHTML génériques sont ainsi partie intégrante du catalogue. Au niveau plus conceptuel, des éléments concrets apparaissent également, par le biais d'un ensemble de formes Microsoft Visio directement utilisables pour composer des prototypes en fil de fer. Ce catalogue, au sein de SmalS, est nommé « StyleGuide » ; les patrons qui le composent sont nommés « *guidelines*[1] ». Une capture d'écran de la page d'accueil de la version 1.7 est présentée sur la Figure 50.

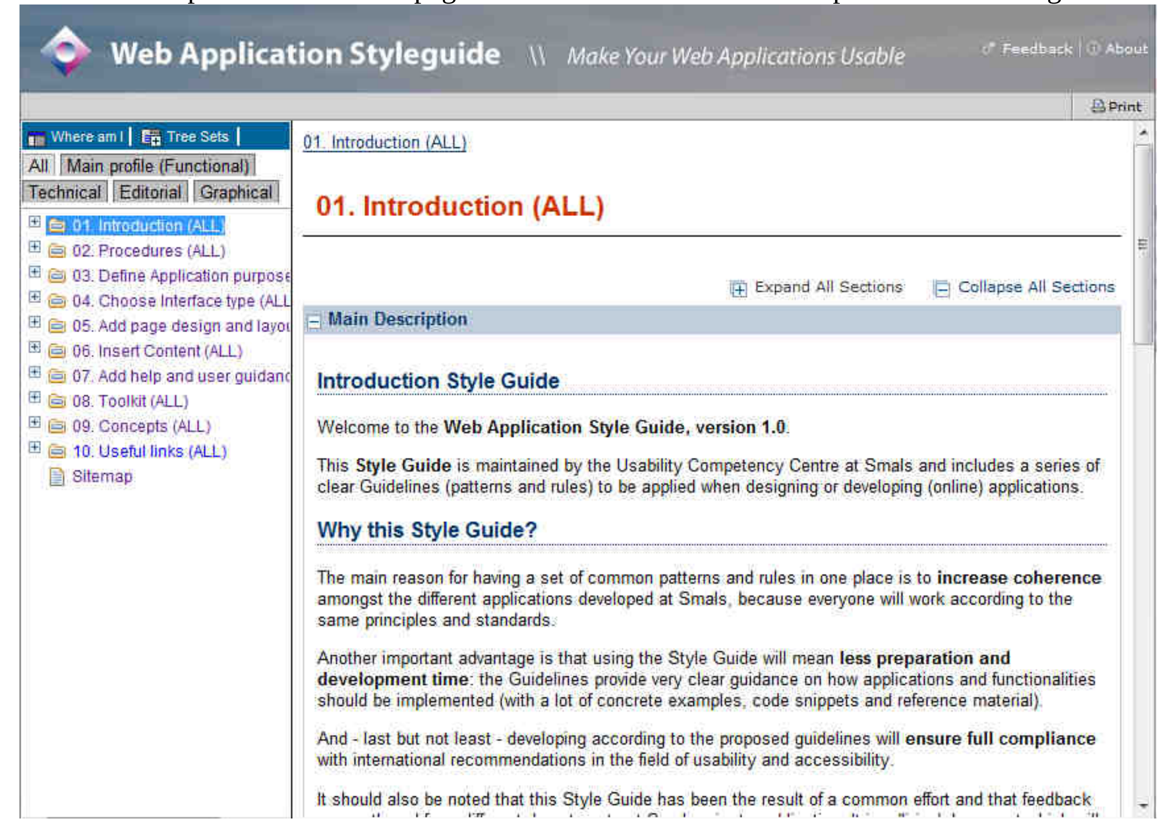


Figure 50. StyleGuide version 1.7

- Pour nous, formaliser le cadre méthodologique du catalogue de patrons : notre effort s'est concentré sur la formalisation du cadre méthodologique du catalogue, et la formalisation de leur contenu. Comme il a été détaillé dans le chapitre 4 « Méthode de conception basée sur des patrons », nous avons étudié les moyens de rendre le catalogue utile dès les premières étapes de la conception d'interface en fournissant des représentations non ambiguës de l'interface quoique partielles, et utilisable (pour les activités d'exploration, et de personnalisation) par tout membre de l'équipe de conception pour améliorer la communication entre intervenants.

5.1.1. Objet de l'évaluation : le StyleGuide

Notre méthode suit les mêmes préceptes de conception d'interface et c'est là le majeur point commun de ces deux approches. Le domaine d'application de ces méthodes est le même,

[1] Indications

l’e-Procuration, ce qui a son importance lorsqu’il s’agit de baser une méthode sur des patrons. Les patrons émergent en effet de l’existant et de la connaissance généralement empirique que les concepteurs ont des solutions de conception déjà éprouvées. Les utilisateurs ciblés sont similaires (à l’exception des développeurs d’interface, puisque nous excluons les recommandations d’un niveau technique) et l’objectif est, dans les deux cas, de les aider dans la réalisation de prototypes moyenne fidélité en début de conception. Le support pour la diffusion des patrons (nommés *guidelines* dans la méthodologie SmalS) est identique : il s’agit d’un site Web. Les structures choisies sont de même nature : il s’agit d’associer les patrons par granularité, et d’offrir ainsi un ordre de consultation qui suive la réflexion naturelle de conception.

Toutefois, dans notre catalogue, la forme des patrons est plus systématisée que dans le StyleGuide : les sections de description de nos patrons suivent une forme immuable, quelque soit leur niveau de granularité. Tandis que SmalS a développé des *guidelines* destinés aux développeurs d’interface, nous avons cherché, de notre côté, à faire la distinction entre prototypes de page et prototypes d’enchaînement d’écrans, formalisés avec StateWebCharts (cf. §4.2). L’opérationnalisation des patrons a ainsi pris des directions différentes : tandis que SmalS s’est attaché à supporter le prototypage de pages (mise à disposition de formes Visio et de fragments de code XHTML), nous avons produit un outil guidant la conception de la structure hypertexte de l’application (cf. chapitre 6). Les différences de fond peuvent être résumées de la façon suivante : tandis que nous avons nourri la spécification d’interface de haut niveau en formalisant et opérationnalisant des patrons d’enchaînements d’écrans, le StyleGuide a approfondi la concrétisation en ajoutant des *guidelines* techniques.

L’écart le plus notable, toutefois, entre le StyleGuide et notre catalogue, est que le StyleGuide est d’ores et déjà utilisé en conditions réelles. L’opportunité d’observer et de nous entretenir avec les utilisateurs du StyleGuide était un moyen d’ouvrir des pistes pertinentes d’amélioration pour notre méthode. En retour de cette opportunité, un rapport d’utilisabilité avec des propositions d’évolution serait remis à SmalS.

5.1.2. Objectifs

Une évaluation du StyleGuide a été mise en place dans le but de valider les hypothèses de nos recherches tout en fournissant un retour d’expérience à SmalS sur les conséquences du déploiement du StyleGuide. L’objectif principal est l’amélioration du StyleGuide tel qu’il existe, tirant de ce fait des leçons pour un déploiement optimal de notre propre méthode. L’évaluation de l’utilisabilité du StyleGuide et le repérage éventuel de problèmes d’utilisabilité permettent ainsi, dans un premier temps, de corriger des erreurs pour améliorer les conditions d’usage des utilisateurs.

Le StyleGuide doit d’abord fournir toutes les qualités de bonne utilisabilité requises en tant qu’application Web. Si l’on applique la définition de l’utilisabilité de Jakob Nielsen (Nielsen 1993) au cas particulier d'une application Web, l'utilisabilité est déterminée par la satisfaction de l'utilisateur, la facilité d'apprentissage, la capacité pour le visiteur de se souvenir d'une organisation et de ses fonctionnalités, l'efficacité, la performance, et la probabilité d'erreurs lors de la réalisation des tâches prévues par le site. Les tâches réalisées par le biais du StyleGuide étant des activités de travail quotidiennes, l’adéquation du StyleGuide à leurs habitudes de travail est primordiale. L’acceptabilité d’une telle méthode doit être évaluée également : toute méthode de travail, à son déploiement, nécessite une phase d’adaptation et requiert des efforts de la part de l’utilisateur. Ces efforts sont-ils récompensés de façon intrinsèque (p.ex. accélération de la conception de *wireframes*) et extrinsèque (p.ex. sentiment gratifiant de voir ses expériences partagées via le StyleGuide) ?

Quoique des remarques puissent être recueillies à ce sujet de façon annexe, le contenu du StyleGuide lui-même ne sera pas évalué. En effet, notre travail porte sur les moyens de mise à disposition de connaissance ergonomique pour la conception d’interface d’e-Procuration, pas sur la connaissance ergonomique elle-même. En revanche, nous profitons de la rencontre avec des concepteurs d’interface pour évaluer l’utilité et l’utilisabilité du formalisme StateWebCharts

et discuter avec eux de la place de la modélisation de la structure hypertexte dans leurs activités de conception. Le possible positionnement de StateWebCharts comme plateforme commune de communication au cours de la conception est discuté également.

5.1.3. Méthodes utilisées

Le StyleGuide a été utilisé par une trentaine de personnes depuis fin 2007, dans le cadre de projets pilotes puis de projets classiques. Il en est toutefois encore à un stade où le contenu aussi bien que la forme évoluent énormément ; une version stable dans sa forme aura été déployée pour la fin de l'été 2008. L'instabilité du StyleGuide nous a fait écarter une inspection heuristique en profondeur. C'est toutefois par le biais d'une inspection informelle du StyleGuide que nous avons abordé l'évaluation : certains problèmes d'utilisabilité ont été relevés, qui ont guidé la formulation des scénarios de l'évaluation.

Une analyse statistique de données d'usage ne convenait pas à ce système encore jeune : la majorité des activités relevées auraient été des activités d'exploration, d'apprivoisement de l'outil. Peu de personnes ayant utilisé le StyleGuide dans leur activité réelle de travail, les données statistiques n'auraient pu être exploitées. De plus, plusieurs versions du StyleGuide sont utilisées en parallèle, ce qui empêche de relever des statistiques d'usage pertinentes. Nous avons toutefois mis en place un questionnaire portant sur des données démographiques, pour sélectionner les sujets que nous rencontrerions. Vu le faible nombre de réponses, tous les sujets ont été évalués.

Toutefois, le StyleGuide a atteint une maturité et un volume de contenu suffisants pour que des évaluations basées sur la participation des utilisateurs puissent être menées de façon pertinente et enrichissante. En avril 2008, avec l'aimable accord et la participation de SmalS, nous avons eu l'opportunité d'effectuer une évaluation du StyleGuide par ses utilisateurs. Nous avons pu mener des observations de faits provoqués ainsi que des entretiens avec des utilisateurs réels du StyleGuide, analystes et chefs de projet. Mener l'évaluation sur leur lieu de travail permettait de réunir un maximum de sujets. Leur temps de travail étant fléché sur des budgets spécifiques, l'évaluation ne devait pas dépasser quatre vingt dix minutes : se rendre sur leur lieu de travail leur évitait de se déplacer. L'évaluation ne se déroulait pas sur le poste de travail du sujet mais dans une salle de réunion équipée d'un ordinateur relié à Internet. Un ordinateur portable et un dictaphone numériques étaient amenés sur place. Aucun matériel ne requerrait ainsi que l'évaluation se déroule en laboratoire.

5.2. MODE DE FONCTIONNEMENT DU STYLEGUIDE

Le StyleGuide vient d'une volonté de standardisation de la conception chez SmalS. Cette standardisation, en plus de représenter un gain de temps et de moyens, assure une conception efficace pour des problèmes récurrents, problèmes de conception d'interfaces Web dans le cas du StyleGuide. Une base de connaissance standard est accessible aux concepteurs, et cette base s'enrichit au cours du temps, à partir des suggestions des concepteurs eux-mêmes, et sous le contrôle d'un comité de direction. De plus, SmalS guide les concepteurs dans l'application de ces standards en fournissant des outils d'opérationnalisation, permettant un prototypage rapide.

Par le biais du StyleGuide, nous menons une évaluation de l'utilisabilité d'une méthode proche de la nôtre, supportée par le StyleGuide lui-même et par des moyens d'opérationnalisation de celui-ci pour les analystes et les développeurs d'interface. Cette section détaille les moyens de diffusion et de maintenance de cette méthode et des supports qui lui sont associés, dans le cadre de notre expérimentation : l'entreprise SmalS.

5.2.1. Utilisateurs du StyleGuide et responsables de sa gestion

La gestion du contenu du StyleGuide doit être formalisée et organisée avec soin pour que son évolution soit souple et que ce contenu soit toujours valide. Le ***Usability Competency Center***[1] est responsable de la maintenance du contenu, et en particulier des recommandations qui découlent de leurs compétences en facteurs humains et en conception d'interface. Ce groupe produit aussi un ensemble de formes Visio qui accompagne le StyleGuide pour la conception de *wireframes* (cf. §5.2.3). Ils fournissent également un support aux concepteurs dans l'utilisation du StyleGuide. Un **comité technique** travaille quant à lui sur la conception et la maintenance des fragments de code XHTML diffusés par le biais du StyleGuide pour la conception de maquettes interactives (cf. §5.2.3). Un **comité de direction** articule ces composantes et valide les *guidelines* avant publication dans le StyleGuide : il est composé d'experts en utilisabilité, de représentants du comité technique et d'experts en analyse fonctionnelle.

Les utilisateurs du StyleGuide sont principalement des analystes fonctionnels et des développeurs d'interface puis, plus occasionnellement des graphistes et des responsables éditoriaux. Le comité de direction est ici considéré comme un utilisateur de la méthodologie du StyleGuide, quoique les tâches qui les concernent soient des tâches de maintenance du StyleGuide. Les tâches d'utilisation du StyleGuide sont listées ci-dessous et résumées dans le Tableau 6 qui donne, pour chaque catégorie d'utilisateurs, la fréquence à laquelle ils réalisent cette tâche.

Tableau 6. Matrice utilisateur / tâches autour du StyleGuide

		Analyste	Développeur d'interface	Responsable éditorial	Graphiste	Comité de direction du StyleGuide
Consulter		++++	+++	++	++	++
Concevoir	Wireframe	++++	-	++	-	-
	Fragment de code	-	++++	-	+++	-
Contribuer	Commentaire	++	++	++	++	-
	Proposer	+	+	-	-	-
	Mettre à jour	-	-	-	-	++++

LÉGENDE fréquences auxquelles un utilisateur réalise une tâche relative au StyleGuide
++++ Très souvent +++ Souvent ++ Parfois + Rarement - Jamais

- **Consultation :** l'utilisateur rencontre un problème de conception d'interface, ou souhaite se référer à un *guideline* dont un de ses interlocuteurs lui a parlé : il cherche le *guideline* dans le StyleGuide. Ce *guideline* peut appartenir à une catégorie de granularité plus ou moins haute en fonction du moment de la conception. Il peut concerner le profil de l'utilisateur ou un autre profil.
- **Conception :** quand l'utilisateur a trouvé un *guideline* qui lui semble pertinent, il tente de l'adapter au projet en cours, afin d'explorer les solutions à son problème d'interface.
  - Adapter **un *wireframe* de page** : un analyste conçoit un *wireframe* de page, sur la base d'éléments proposés dans le StyleGuide. Il renseigne les en-tête et pied de page, alimente le contenu avec les informations du projet ;
  - Adapter **du code existant** : un développeur d'interface ou un graphiste personnalise les fragments de code apparaissant dans le StyleGuide pour concevoir une maquette interactive ;
- **Contributions** : le StyleGuide est voué à évoluer dès que de nouveaux problèmes de conception émergent, repérés par ses utilisateurs eux-mêmes. Ces derniers proposent de modifier des *guidelines* existants, ou d'en ajouter de nouveaux, le processus de sélection, de formalisation et de publication dans le StyleGuide étant réservé au comité de direction du StyleGuide.
  - **Commenter** un *guideline* : un utilisateur envoie un courrier électronique pour commenter un *guideline* existant, en vue de l'améliorer ;

[1] Centre de compétences en utilisabilité

– **Proposer** un nouveau *guideline* : un utilisateur propose un nouveau *guideline*, incluant un *wireframe*, sur la base de son expérience sur le terrain ;
– **Mettre à jour** le StyleGuide : le comité de direction du StyleGuide ajoute un nouveau *guideline* ou modifie un *guideline* existant, sur la base de ses réflexions, de commentaires ou de propositions des utilisateurs.

5.2.2. Contenu et organisation du StyleGuide

Le StyleGuide prend la forme d'une application Web conçue avec Eclipse. Il est écrit en Anglais, comme plateforme linguistique commune aux Belges francophones, germanophones et néerlandophones, et il est accessible via l'Intranet de l'entreprise. Il est composé de *guidelines* de conception d'interfaces, orientées selon le profil de l'utilisateur, mais toujours adaptées à un projet en début de conception ou de refonte d'interface. La mise en page des *guidelines* permet une prise d'informations rapide pour un utilisateur expert, à la recherche de solutions opérationnelles. Au contraire, l'utilisateur novice peut s'attarder sur les parties argumentant les recommandations données, qu'il soit inexpérimenté en utilisabilité, en conception d'interface, ou simplement nouvel entrant dans l'entreprise et encore peu connaisseur des standards qui y sont utilisés. Des définitions de concepts apparaissent également en tête de tout *guideline* pour rendre cohérent le vocabulaire employé au sein de l'entreprise. Un thésaurus de ces concepts apparaît à partir de la version 2.0 (déploiement prévu en mai 2008).

Plusieurs profils d'utilisateurs se côtoient : les informations des uns ne servent pas nécessairement aux autres, mais cette combinaison de vues permet de faire comprendre à chacun ce que les autres ont de compétences et de besoins. Les analystes étant la cible principale du StyleGuide, la vue les concernant apparaît de façon constante, quelque soit le profil choisi (p.ex. la vue du profil graphiste fait apparaître les recommandations pour les graphistes et pour les analystes). Une vue globale est prévue, mettant à disposition de l'utilisateur du StyleGuide les recommandations proposées à tous les profils décrits ci-après.

- **Analyste fonctionnel :** ce profil contient des informations sur les moyens de réaliser un *wireframe* qui réponde aux besoins fonctionnels et informationnels du client tout en respectant les standards de SmalS et les recommandations ergonomiques. Pour aiguiser la sensibilité à l'utilisabilité de l'analyste, paraissent aussi des liens complémentaires, des erreurs de conception communément rencontrées ainsi que des justifications théoriques des recommandations pratiques ;
- **Développeur :** des recommandations techniques sont diffusées pour ce profil, sous forme de fragments de code à insérer, avec une justification théorique du bien-fondé de la recommandation ainsi que des références à des standards respectés par SmalS, tels que les standards d'accessibilité du W3C (niveau 1) ;
- **Responsable éditorial :** des conseils concernant la façon d'intituler les liens hypertexte, les titres, les labels de formulaire ou encore les en-tête et pieds de page des applications développées par SmalS sont diffusés ici. Ces recommandations sont parfois des standards très directifs (p.ex. présentation des crédits du site), mais peuvent aussi être de simples conseils de rédaction ;
- **Graphiste :** dans ce profil, figurent des directives de positionnement des éléments de l'interface, de coloration de ces éléments, de typographies, et de tout ce qui concerne l'identité visuelle de l'application en cours de conception.

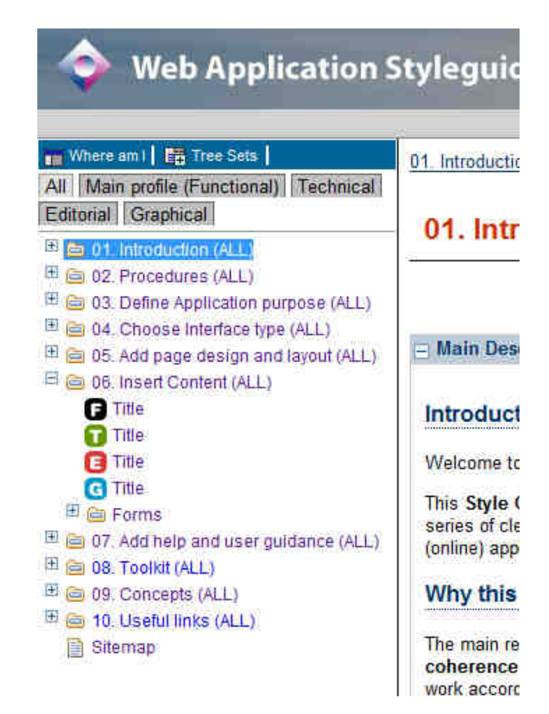


Figure 51. Colonne de navigation du StyleGuide, par catégories de guidelines (vue globale)

Chaque profil d'utilisateur dispose donc de recommandations adaptées à ses activités de conception. Les recommandations du StyleGuide sont groupées par catégories, suivant l'ordre naturel de réflexion pour la conception, c'est-à-dire par ordre de granularité croissant. Ces catégories apparaissent de façon linéaire dans la colonne de navigation du StyleGuide, comme montré sur la Figure 51.

- **Définition du propos de l'application :** le but de cette catégorie est de clairement définir l'objectif de l'application. S'agit-il d'enregistrer des données de la part de l'utilisateur de l'application à concevoir ? De lui en délivrer sous forme d'informations ou de documents ? De lui permettre de consulter et de modifier des informations ?

- **Sélection de l'interface utilisateur adéquate :** les recommandations de cette catégorie concernent la définition de la navigation globale de l'application. Le type de structure hypertexte est défini, c'est-à-dire l'enchaînement d'écrans. L'offre de navigation apparaissant au travers de toutes les pages est également définie ici : elle peut par exemple figurer sous forme de barre de navigation horizontale, ou encore d'arbre de navigation dont les éléments se déploient sur demande de l'utilisateur.

- **Ajout du design de la page et de sa mise en page :** les informations fournies dans cette catégorie concernent la mise en page et la présentation d'éléments permanents, hors de la zone de navigation principale, permettant la navigation au fil des pages de l'application (p.ex. liens apparaissant en en-tête). C'est aussi à ce stade que la mise en page du cœur des pages est définie, en dehors de la zone consacrée à la navigation qui a été définie dans la catégorie précédente.

- **Insertion de contenu relatif à l'interface :** le contenu est ici abordé, avec une grande partie des recommandations dédiée à la conception de formulaires. Leur mise en page est définie ici, ainsi que des conseils d'identification des labels, et des modèles de formulaires classiques (p.ex. saisie d'informations identifiant une entreprise belge).

- **Ajout de l'aide et du guidage de l'utilisateur :** cette catégorie concerne en grande partie la prévention des erreurs de saisie dans les formulaires ou leur signalement lorsqu'une erreur a été détectée après validation. Le guidage passe également par

des pages de validation de procédures, attestant de la bonne conduite de l'application, ou au contraire de retours clairs sur des incidents techniques qui pourraient survenir.

5.2.3. Opérationnalisation du StyleGuide

Le StyleGuide est un recueil de recommandations pour la conception d'interfaces dans le cadre de l'entreprise, pour des problèmes communément rencontrés, recommandations pratiques, concrètes et prêtes à l'emploi. Toutefois, pour constituer un véritable pas dans la démarche de conception suivie chez SmalS, ces recommandations sont accompagnées de modules prêts à être intégrés au travail quotidien des analystes et des développeurs d'interface pour la conception de maquettes en fil de fer réalisées avec Microsoft Visio pour les premiers, et de maquettes interactives en XHTML pour les seconds.

Intégration à Visio

Les analystes disposent de formes Visio standardisées et correspondant aux modules fonctionnels dont ils ont besoin au moment de la réalisation des premières maquettes. Si un *guideline* dispose d'une forme[1] associée dans Visio, alors elle y apparaît sous le même intitulé. L'intitulé du pochoir[2] (ensemble de formes, apparaissant dans la colonne de gauche de la Figure 52) auquel elle appartient est le titre de la catégorie où le *guideline* apparaît dans le StyleGuide. Cette correspondance de terminologie est importante pour guider l'analyste dans le passage du StyleGuide à Visio, c'est-à-dire de la théorie à la pratique. Cet ensemble de formes est accompagné d'icônes et d'images standard, pour rendre réaliste le *wireframe*. Toutefois, le maquettage est ici purement fonctionnel et n'entre pas dans le détail de la présentation. Les formes sont volontairement sobres au niveau graphique, pour que les considérations graphiques n'interfèrent pas avec l'analyse fonctionnelle, en particulier lors de discussions avec les clients. Le design retenu est ainsi en équilibre entre élégance et simplicité, pour avoir une allure professionnelle et réaliste, tout en se gardant de tout choix de présentation.

Cet outil d'opérationnalisation accélère la fabrication de maquettes, et autorise des modifications plus efficaces. Les éléments d'interface présents sont en effet indépendants les uns des autres, et le fond d'écran commun à toutes les pages (avec en-tête et pied de page par exemple) peut être géré à part des contenus. Pour une utilisation optimale, des formations à l'utilisation de Visio et plus particulièrement des formes issues du StyleGuide sont dispensées aux analystes de SmalS. Cette flexibilité est appréciable puisque les demandes de modification ont systématiquement lieu à ce stade de la conception où le client expose puis affine les besoins fonctionnels de son application. Plus l'analyste fait preuve de réactivité dans les modifications qu'il apporte à son *wireframe*, plus les modifications ont de chances d'être validées au fur et à mesure de leur traitement.

[1] En Anglais : *shape*
[2] En Anglais : *stencil*

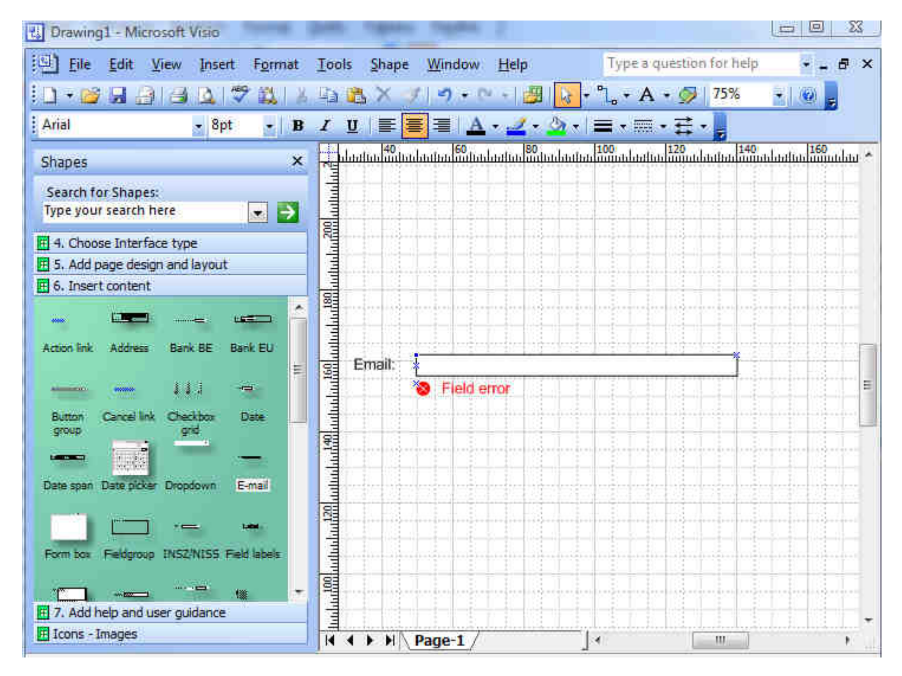


Figure 52. Formes issues du StyleGuide apparaissant dans Microsoft Visio (version 1.7)

Intégration à Eclipse

La seconde catégorie d'utilisateurs jouissant d'une opérationnalisation du StyleGuide (au stade 2.0) est celle des développeurs d'interface. L'une des tâches qui leur est attribuée en début de conception est la réalisation d'une maquette interactive en XHTML, sur la base des prototypes de pages en fil de fer réalisés par les analystes et sur la base de leurs recommandations auxiliaires (p.ex. éléments d'interfaces déclencheurs de la navigation). Cette étape de production d'une maquette interactive permet l'émergence d'éventuels dysfonctionnements dans l'interaction avec l'application ou dans son comportement dynamique qui peuvent être passés inaperçus dans la phase de prototypage de pages. Les considérations graphiques sont également réfléchies à ce stade, et il est question de choix de couleurs de fond, de typographies (c'est-à-dire de la définition des styles CSS), d'identité graphique et de dessin d'icônes appropriées.

Pour soutenir cette étape, en plus des recommandations techniques accessibles aux développeurs dans le StyleGuide, des fragments de code XHTML sont fournis, pour être assemblés les uns aux autres, à la manière des formes Visio. Certains fragments définissent la mise en page dans son ensemble (p.ex. page avec deux colonnes dont la colonne de navigation à gauche), d'autres décrivant des éléments plus précis de l'interface y sont intégrés (p.ex. formulaire de saisie d'une adresse email). Cette activité se déroule sous Eclipse, cadre de développement classique des développeurs d'interface de SmalS. La personnalisation des fragments est manuelle, sur la base de fragments au graphisme sobre, tels que les formes Visio. Une formation est proposée aux développeurs d'interface en même temps que celle proposée aux analystes. La première partie les concerne tous, elle porte sur l'utilisabilité et la conception d'interfaces en général, ainsi que sur le prototypage tel qu'il est vu par chacune de ces catégories d'utilisateurs. Puis quand les travaux pratiques portent sur Visio pour les analystes, les développeurs d'interfaces s'intéressent au maquettage XHTML.

5.3. Définition des hypothèses

Le déploiement du StyleGuide dans l'entreprise SmalS est une illustration concrète de l'introduction d'une méthode de conception basée sur des patrons telle que celle que nous présentons dans cette thèse. La méthode est basée sur la réutilisation d'un projet sur l'autre de solutions à des problèmes de conception d'interfaces : elle est donc supposée raccourcir le temps de résolution des problèmes d'interface et améliorer tant le déroulement du processus de concep-

tion d'interface que son produit. Nous souhaitons vérifier l'hypothèse de recherche selon laquelle la conception de l'interface est une activité qui peut servir de plateforme de communication entre intervenants, tous statuts confondus. Les discussions sur la spécification des besoins puis sur les moyens de les satisfaire seront facilitées par la présence de cette plateforme offrant une compréhension mutuelle des intervenants. Pour Claude Bernard (1865, cité par (Reuchlin 1992)), « La seule preuve qu'un phénomène joue le rôle de cause par rapport à un autre, c'est qu'en supprimant le premier, on fait cesser le second ». C'est à partir de ce raisonnement que nous définissons ci-après des hypothèses de recherche, basées sur des phénomènes sur lesquels il nous est possible d'agir. Ces phénomènes seront provoqués lors d'expérimentations, afin de confirmer ou d'infirmer la validité de nos hypothèses.

5.3.1. Le StyleGuide et un formalisme de description de la structure hypertexte sont utilisables et ont un pouvoir expressif satisfaisant pour la conception d'interface

La solution proposée par un patron issu du StyleGuide est prête à être adaptée au contexte : application de la stratégie de marque ou *branding* (jeu de couleurs, typographie, logo), remplacement des données génériques illustratives par les données réelles de l'application, multiplication des pages du modèle de navigation en fonction du nombre d'étapes à franchir, ajout de pages au modèle de navigation… Cette personnalisation des prototypes en fil de fer est une activité rapide, peu coûteuse et efficace du point de vue de l'avancement de la conception d'interface.

La première hypothèse que nous émettons porte sur l'utilisabilité du StyleGuide, des *guidelines* qu'il contient, ainsi que des formes Visio associées. À cette utilisabilité doit s'ajouter la satisfaction subjective, le sentiment de réaliser un artefact utile pour la conception, servant de support à la communication avec le client et aussi avec l'équipe de conception. Ces supports doivent, de plus, être en adéquation avec les habitudes de travail des utilisateurs de la méthode. Tous ces paramètres entrent en jeu dans l'acceptabilité et l'adhésion au StyleGuide.

Hyp. 1

Si un utilisateur doit produire des *wireframes* de page pour un site Web
avec l'aide du StyleGuide

Alors il est capable de les trouver dans le StyleGuide
et de les adapter avec satisfaction
et en ayant le sentiment de fournir une documentation utile pour la spécification de l'interface.

L'observation de faits provoqués nous permettra d'observer la manière dont les utilisateurs se servent du StyleGuide. Le scénario provoqué pour l'observation portera sur la conception de *wireframes* de page avec le support du StyleGuide et également des formes Visio fournies par SmalS. Les facteurs observés seront les suivants :

- Le niveau d'utilisation du StyleGuide : combien de *guidelines* et de pages du StyleGuide (p.ex. table des matières d'une catégorie, définition d'un concept) consultés ? La pertinence des *guidelines* n'est pas considérée, dans la mesure où le contenu du StyleGuide n'est pas évalué ;
- Le niveau d'appropriation des formes Visio : l'utilisateur personnalise-t-il l'environnement du logiciel Visio tel qu'il lui est présenté ?
- La satisfaction à l'utilisation du StyleGuide et des formes Visio ;
- Les habitudes en matière de conception de *wireframes* de pages, et l'adéquation de la méthode à celles-ci ;

- Le niveau d'adhésion à la méthodologie autour du StyleGuide : enthousiasme, compréhension des préceptes de la méthode, sensation d'être concerné.

De plus, nous émettons l'hypothèse que si les *guidelines* d'enchaînement d'écrans étaient modélisés systématiquement avec un formalisme de description de la structure hypertexte, ils offriraient un support efficace et utilisable à la communication autour de problèmes de conception d'interfaces. Nous évaluons cette hypothèse en utilisant le formalisme StateWebCharts (Winckler et Palanque 2003) pour décrire ici la structure hypertexte.

Hyp. 1bis

Si un utilisateur doit produire des modèles de structure hypertexte
pour un site Web avec l'aide du formalisme StateWebCharts
et qu'un exemple lui a été décrit en détails

Alors il est capable de le produire
et en ayant le sentiment de fournir une documentation utile pour la spécification de l'interface.

Pour évaluer cette seconde hypothèse, l'observation portera sur la conception d'un modèle StateWebCharts, sur le même cas d'étude que la première hypothèse. Conserver le même cas d'un scénario à l'autre permettra d'évoquer de façon complète les étapes de la spécification de l'interface. Le réalisme du scénario donnera encore à l'utilisateur l'occasion de faire des ponts avec son expérience quotidienne et d'en faire part à l'évaluateur. Des entretiens permettront de collecter des informations complémentaires, notamment quant à la satisfaction subjective ressentie par l'utilisateur au cours de la réalisation des scénarios. Les facteurs observés seront les suivants :

- La connaissance du formalisme StateCharts ;
- Les habitudes en matière de modélisation de la structure hypertexte d'une application Web : outils et formalismes utilisés, informations représentées, destinataires de ce modèle ;
- L'utilisation de tous les aspects du formalisme présenté dans l'exemple introductif : état parent, état initial, état final, apposition de contraintes sur les transitions, état composé, contenu de page dynamique versus statique ;
- L'avis subjectif du sujet quant à StateWebCharts : informations manquantes, lisibilité et potentiel pour être montré au client, complétude et potentiel pour servir de support aux développeurs d'interface.

5.3.2. Les *wireframes* de page et le modèle de structure hypertexte aident au consensus

Dans un projet de conception d'application d'e-Gouvernement, de nombreux intervenants se croisent, peinent à communiquer faute d'un langage commun, et peinent à prendre certaines décisions de conception faute d'arguments qui puissent être entendus par tous, toutes exigences et tous intérêts confondus. Ce type de blocage intervient régulièrement dans ce contexte où les intérêts politiques ont un poids important. La décision permettant le consensus entre intervenants ne doit pas aller à l'encontre des intérêts de l'utilisateur de l'application, pourtant, il arrive que ce soit le cas. Pour soutenir les arguments des experts en conception d'interface (développeur d'interfaces, consultant en utilisabilité), le recours à une solution attestée comme valide par expérience, c'est-à-dire à un patron de conception d'interface, peut représenter une issue à de tels blocages décisionnels.

Nous émettons ici l'hypothèse selon laquelle les *wireframes* de page couplés au modèle de la structure hypertexte de l'application en cours de conception aident à l'élaboration d'un

consensus. Ils sont lisibles et critiquables et, sur leur base, une solution consensuelle peut être trouvée et spécifiée.

Hyp. 2

Si un utilisateur U2 est mis en présence de *wireframes* d'enchaînement d'écrans et de *wireframes* de pages produits par un autre utilisateur U1

Alors U2 peut lire les *wireframes* de U1
et U2 peut comparer sa propre production à celle de U1
et U2 peut proposer des améliorations pour sa production et/ou pour celle de U1

La méthode expérimentale qui nous permettra de mesurer la validité de cette hypothèse consistera en la présentation au sujet de *wireframes* de page et d'un modèle StateWebCharts à commenter. Ces artefacts traitent du cas d'étude sur lequel il aura eu à réfléchir précédemment. Il lui sera demandé de lire et de critiquer ces propositions, en vue d'en tirer le meilleur, et d'envisager, oralement, de combiner ces atouts à ceux de ses propres propositions. Les facteurs observés seront les suivants :

- Lisibilité et expressivité du modèle StateWebCharts étalon ;
- Niveau de culture en utilisabilité, dans le cadre de la critique des *wireframes* de pages ;
- Recul et souplesse dans la recherche de consensus : cette variable sera considérée avec circonspection dans la mesure où elle est fortement dépendante du profil du sujet concerné, de son expérience et de sa personnalité.

5.3.3. Le StyleGuide favorise la réutilisation de la connaissance issue d'expériences de conception

Lorsqu'un problème de conception d'interface est rencontré, l'équipe de conception se réfère au StyleGuide pour connaître la solution adaptée qui est préconisée au vu de l'expérience de conception de SmalS. Lorsque le problème rencontré n'apparaît pas dans le StyleGuide, ou lorsque la solution proposée n'est pas adaptée au contexte du projet en cours, l'équipe de conception transmet ses commentaires à l'équipe de gestion du StyleGuide. Un cadre d'édition des patrons formalisé de façon fortement guidée rendrait les propositions d'ajout ou de modification efficace, ciblées sur une ou plusieurs des propriétés d'un patron. Le StyleGuide offre un cadre à la fois formel et évolutif pour le stockage, la maintenance et l'accès aux connaissances acquises lors des projets d'e-Gouvernement de SmalS.

Nous émettons ici l'hypothèse selon laquelle un utilisateur confronté à un problème non référencé est capable de formuler une suggestion de mise à jour du StyleGuide. Nous supposons que cette activité de suggestion a un impact positif sur le sentiment d'implication de l'utilisateur vis-à-vis de l'évolution du StyleGuide.

Hyp. 3

Si un utilisateur est confronté à un problème de conception d'interface qui n'est pas référencé dans le StyleGuide
et un canevas de formalisation de patron est disponible pour commentaires sur le StyleGuide

Alors l'utilisateur peut formaliser et soumettre un nouveau patron en un temps jugé raisonnable par l'utilisateur lui-même.

L'observation de faits provoqués servira de méthode de validation de cette hypothèse, encore une fois en combinaison avec un entretien complémentaire. Une fonctionnalité doit être ajoutée au site qui a servi d'étude de cas pour les scénarios précédents. Cette fonctionnalité implique notamment l'ajout d'une page ; pour cette page, aucune recommandation ne figure dans le StyleGuide. Pour que la totalité de l'expérimentation ne dépasse pas 90 minutes, cette absence de recommandation sera signalée au sujet, et nous lui demanderons comment il se serait rendu compte de l'absence d'un *guideline* adapté à son cas. L'alternative plus coûteuse en temps aurait été d'observer le sujet à la recherche de ce *guideline* particulier.

En situation réelle, le sujet aurait alors à utiliser d'autres ressources que le StyleGuide pour produire un *wireframe* de page pertinent : pourquoi ne pas partager les conclusions de cette expérience par le biais du StyleGuide ? Via un formulaire papier, le sujet devra soumettre une suggestion de *guideline* à l'équipe responsable du StyleGuide, à propos de cette page absente du StyleGuide. Les facteurs observés seront les suivants :

- Indices de l'absence d'un *guideline* adapté : à quel moment le sujet sait-il qu'un *guideline* est absent du StyleGuide ? Quelle est alors son attitude ? ;
- Ressources utilisées pour la composition de la suggestion : l'utilisateur cherche-t-il des exemples, des recommandations techniques, des recommandations ergonomiques ? ;
- Aisance dans le remplissage du formulaire ;
- Sentiment de gratification ressenti pour la contribution au StyleGuide ;
- Degré d'implication dans l'évolution du StyleGuide ;
- Motivation personnelle et sentiment quant à la faisabilité concrète de telles contributions à l'évolution du StyleGuide.

5.4. ÉTUDE ÉVALUATIVE

Des évaluations ont été menées pour confronter les hypothèses émises aux comportements d'usage des utilisateurs du StyleGuide. Les données recueillies ont été dans un premier temps quantitatives, via un questionnaire en ligne. Puis des observations de réalisation de scénarios sont intervenues, ponctuées d'entretiens dirigés pour approfondir des points soulevés durant la réalisation du scénario. La démarche expérimentale suivie est décrite dans cette section, c'est-à-dire le matériel employé, l'environnement qui a accueilli les évaluations, les personnes impliquées et les scénarios qui ont été observés.

5.4.1. Utilisateurs et contexte de l'évaluation

Le premier pas de cette expérimentation est la mise en place d'un pré questionnaire, réalisé et diffusé grâce au site Web SurveyMonkey[1] (cf. Annexe C) sur les mois de mars et avril 2008. Le but de ce pré questionnaire est de recueillir des données démographiques pour mieux cerner le profil des utilisateurs, et pouvoir sélectionner, parmi eux, un échantillon représentatif de sujets avec lesquels mener l'expérimentation. Finalement, compte tenu du faible nombre de participants, tous seront retenus, et certains participeront à l'expérimentation sans avoir au préalable rempli le pré questionnaire. Onze questionnaires ont, au total, été complétés, remplis par les sujets rencontrés pour l'expérimentation, mais aussi par des personnes n'ayant pas pu participer à l'expérimentation. Aucune étude statistique n'a été menée sur ces données trop peu nombreuses ; une interprétation réservée des résultats a toutefois pu être fournie (cf. §5.5.1).

Ensuite, les observations de faits provoqués ont été menées en avril 2008, dans le but de confirmer ou infirmer les hypothèses émises dans la section 5.3. Ils ont été menés en moyenne durant 90 minutes, avec des sujets analystes. Certains scénarios provoqués (cf. §5.4.4) incluant

[1] http://www.surveymonkey.com (dernier accès : 25 sept. 08)

l'utilisation du StyleGuide et des outils associés, leur observation nous permet de relever des problèmes récurrents. Les sujets sont encouragés à verbaliser leurs réflexions et leurs actions (protocole *think aloud*), et interrompent parfois leurs scénarios pour une digression sur leurs expériences avec le StyleGuide, ou en tant qu'analyste. Des entretiens consécutifs aux scénarios permettent de recueillir des informations complémentaires (cf. Annexe D). Les expérimentations ont été intégralement enregistrées au format audio, les activités avec le StyleGuide et Visio également, avec l'accord de consentement éclairé des sujets. Huit sujets ont pu être rencontrés, ce qui est de nouveau un nombre trop faible pour pouvoir envisager un rapport statistique sur leurs opinions et leurs activités. Toutefois, comme mis en avant par Jakob Nielsen (Nielsen et Landauer 1993), ce nombre de sujets est optimal pour un repérage des principaux problèmes d'utilisabilité.

5.4.2. Pré questionnaire

Un questionnaire destiné au recueil de données démographiques pour la sélection des sujets à rencontrer en entretien, a été diffusé aux utilisateurs du StyleGuide au cours du mois de février. Pour cela, un courrier électronique leur a été envoyé par nos contacts au centre d'expertise en utilisabilité de chez SmalS, avec l'adresse URL du questionnaire à remplir (cf. Annexe C). Il y était spécifié qu'un entretien complémentaire aurait éventuellement lieu dans le courant du mois d'avril 2008. Ces entretiens ont finalement eu lieu du 15 au 18 avril 2008. L'intervention d'un intermédiaire permettait de signifier aux utilisateurs que cette évaluation entrait dans le cadre de leur travail quotidien, et que le temps imparti à sa réalisation serait comptabilisé dans leurs heures de travail. L'observation de réalisation d'activités quotidiennes vulnérabilise en effet le sujet, qui peut craindre que l'évaluation soit en réalité une inspection de ses aptitudes de travail.

Le pré questionnaire a été rempli par une population de N=11, principalement des analystes (10 sur 11, le 11ème participant étant un chef de projet ayant participé, en tant qu'analyste, aux premières phases de lancement du StyleGuide). Avant les entretiens menés in situ, 4 questionnaires avaient été remplis. Durant les entretiens, ceux des sujets qui ne l'avaient pas rempli ont été encouragés à le remplir et l'ont effectivement fait (4). Suite aux entretiens, un email était envoyé aux participants pour les remercier de leur implication ; il leur était alors suggéré de proposer à leurs collègues utilisateurs du StyleGuide de remplir le questionnaire, pour qu'un volume de données intéressant puisse être traité. 3 questionnaires supplémentaires ont alors été remplis.

5.4.3. Matériel de l'évaluation

Chaque participant est convoqué sur l'un des trois sites SmalS de Bruxelles par un responsable du centre d'expertise en utilisabilité. Les participants sont des analystes ou ont été analystes dans leur parcours professionnel. L'accueil du sujet consiste en une présentation des grandes lignes de ce travail de thèse et en particulier les rapports entre celui-ci et le StyleGuide. Le séjour chez SmalS d'octobre 2006 est mentionné également, ainsi que le nom de l'hôte de ce séjour que quelques sujets connaissent pour avoir participé à des projets pilote avec le StyleGuide. Ce séjour est présenté comme contribution aux recherches préliminaires de la conception du StyleGuide. Les supports utilisés pour la réalisation des scénarios sont reproduits dans l'Annexe D.

Un dictaphone enregistre au format MP3 l'ensemble de l'évaluation, discussions annexes, exécution de scénarios et entretiens complémentaires. Un ordinateur portable est à la disposition du sujet pour réaliser les scénarios requerrant un enregistrement de captures vidéo d'écrans (réalisées grâce à la version de démonstration du logiciel SnagIt[1]). Après signature de l'accord de consentement éclairé (voir Annexe D) stipulant ces conditions d'enregistrement de l'évaluation, la discussion et les interactions de l'utilisateur avec l'ordinateur portable sont ainsi tracées, et

[1] http://www.techsmith.fr/snagit.asp (dernier accès : 25 sept. 08)

l’évaluateur peut se concentrer sur la conduite de la discussion. Microsoft Visio 2003 (en Anglais) et les formes issues du StyleGuide version 1.7, ainsi que le StyleGuide version 1.7 lui-même (affiché dans le navigateur Web Firefox) sont disponibles sur cet ordinateur portable. Par mesure de sécurité, les ordinateurs portables externes ne sont pas autorisés à se connecter sur le réseau de l’entreprise : un poste de travail fixe permet toutefois, dans la majorité des cas, une connexion à Internet. La Figure 53 présente un cas typique de mise en place pour les évaluations. L’évaluateur est placé à droite de la photo, ce qui lui permet de prendre des notes, d’observer le sujet évalué et de manipuler l’ordinateur portable, notamment pour le lancement et l’arrêt des captures vidéo d’écran. Les scénarios à réaliser font toujours l’objet d’un descriptif imprimé sur papier. Cette impression est placée devant le sujet qui en dispose à sa guise. Les entretiens complémentaires sont remplis sur papier également, directement par le sujet.

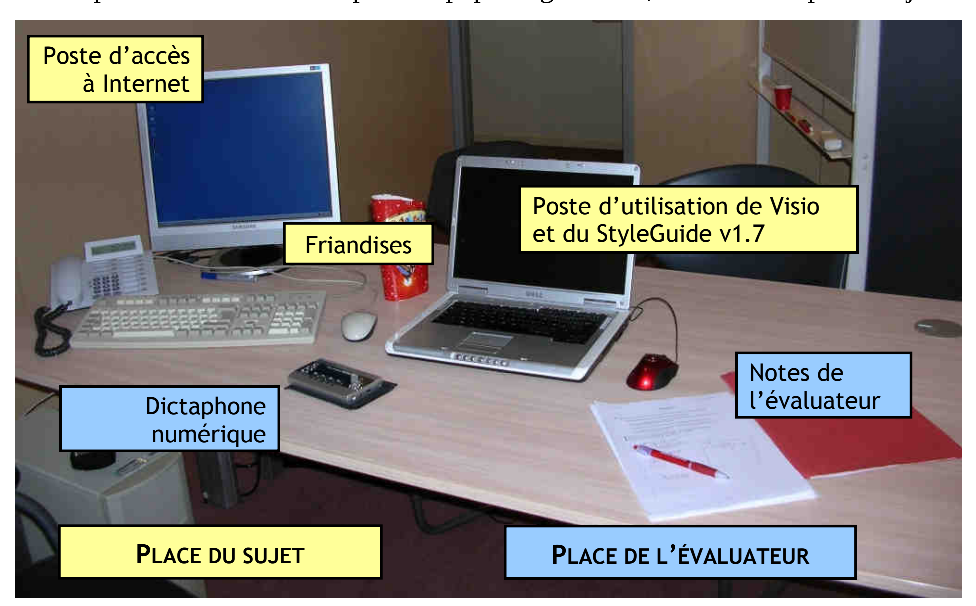


Figure 53. Exemple typique d’environnement mis en place pour les évaluations

5.4.4. Scénarios observés

L’évaluation est présentée comme une occasion d’observer les comportements d’usage du StyleGuide, et d’en déduire des pistes à encourager et d’autres à corriger. Un entretien préliminaire (voir Annexe D) vise à lancer la discussion sur des questions vagues à propos de l’intégration du StyleGuide dans les projets, ce qui donne l’occasion au sujet de préciser son profil, et de donner un avis spontané sur son utilisation du StyleGuide. Puis, des scénarios d’exécution de scénarios sont observés afin de comprendre plus objectivement les modes d’usage et d’évaluer la validité des hypothèses de recherche établies dans la section 5.3. Les deux premiers scénarios portent ainsi sur l’utilisabilité et l’utilité du StyleGuide pour la conception de *wireframe* de pages et du formalisme StateWebCharts pour la spécification de la navigation. Le troisième scénario concerne la lisibilité des productions des deux tâches précédentes, et la possibilité de discussion autour de tels artefacts, à la recherche de consensus. Le quatrième et dernier scénario porte sur l’évolutivité du StyleGuide et le rôle que ses utilisateurs pensent y jouer. Un entretien final permet ensuite au sujet de préciser des points s’il le souhaite.

Les scénarios ont été observés sur une population de N=8 sujets, principalement des analystes (5 sur 8), mais aussi un architecte et deux chefs de projet, ces deux derniers n’étant pas des utilisateurs directs du StyleGuide, mais des incitateurs à son usage.

Scénario 1. Concevoir un* wireframe *de page avec le support du StyleGuide

Le site Web d’une cantine communale doit être conçu. Un accès par identifiant autorise un parent à consulter, ajouter ou supprimer des repas pour ses enfants. Un autre accès par identifiant permet au personnel de la cantine de consulter le nombre de repas prévus à une date don-

née. Le sujet doit réaliser un *wireframe* de la page qui permet à un parent de consulter, ajouter ou supprimer des repas pour ses enfants. Pour ce faire, le StyleGuide (version 1.7) est mis à la disposition du sujet, ainsi que les formes correspondantes, ouvertes dans Visio comme indiqué sur Figure 54. Il est signalé au sujet qu'il est libre de personnaliser cet environnement à sa guise.

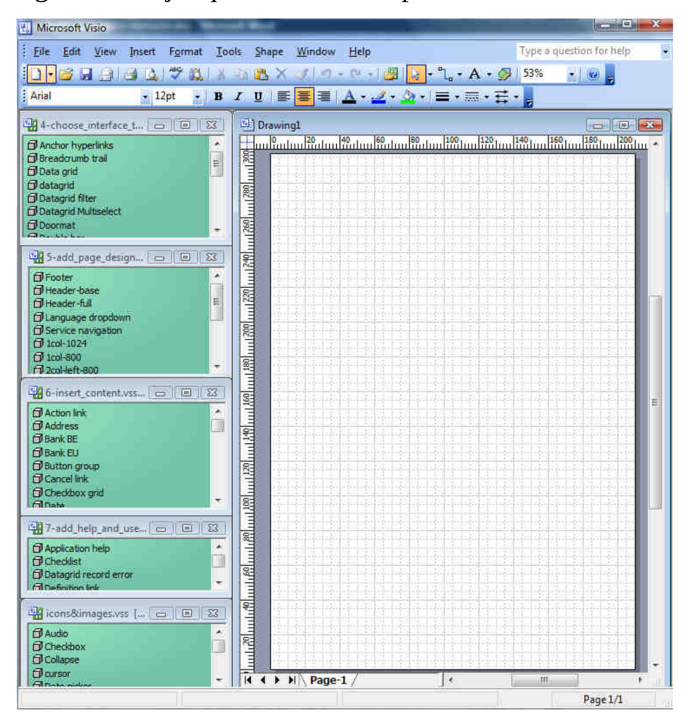


Figure 54. Environnement de travail par défaut pour la réalisation d'un *wireframe* avec Visio

Ce scénario est mis en place dans le but de valider l'hypothèse 1 (cf. §5.3.1) qui concerne l'utilisabilité du StyleGuide, de ses *guidelines* et des formes Visio associées. Il donne l'opportunité d'observer si des habitudes de travail existent avec Visio et les formes du Style-Guide, auquel cas l'environnement de travail sera personnalisé. L'alternance de consultation du StyleGuide et de la fabrication concrète du *wireframe* de page est notée également. Des renseignements plus généraux sont également pris sur les habitudes de travail des analystes pour la conception de prototypes : commencent-ils par penser l'application dans sa globalité ou bien se concentrent-ils sur son cœur, c'est-à-dire sur la partie fonctionnelle essentielle ?

Scénario 1bis. Concevoir une structure hypertexte de site avec StateWebCharts

Après avoir réfléchi aux pages de l'application, la structure hypertexte du site Web de la cantine dans son intégralité est maintenant appréhendée avec, comme contrainte, l'utilisation du formalisme de modélisation de la navigation StateWebCharts. Une partie seulement du formalisme est présenté, celui qui concerne les interactions avec l'utilisateur, laissant de côté les actions du serveur. Des explications sur la syntaxe du formalisme sont données sur deux exemples simples (cf. Annexe D). Le modèle StateWebCharts est réalisé sur papier.

Les observations menées sur ce scénario visent à valider l'hypothèse 1bis (cf. §5.3.1), c'est-à-dire à évaluer la facilité et la satisfaction avec laquelle le sujet s'approprie et utilise le formalisme de modélisation de la structure hypertexte, ici StateWebCharts. Le nœud de la navigation est ici l'identification du visiteur : s'agit-il d'un parent ou d'un membre du personnel de la cantine ? Quelle navigation proposer à chacun de ces profils ? Il est également intéressant d'observer la manière dont le sujet découpe les pages, la manière dont il appose des labels sur les états et les transitions. À l'issue de la réalisation du modèle, un entretien porte sur la satisfaction subjective éprouvée lors de l'utilisation de ce formalisme et les difficultés rencontrées. Le pouvoir expressif de la partie du formalisme présenté est également mis en cause. Si on considère le *wireframe* réalisé lors du premier scénario, et en imaginant que les autres pages ont été

spécifiées de même et en un temps plus réaliste, si on considère aussi le modèle StateWebCharts réalisé : des éléments manquent-ils, selon lui à la spécification de l'interface de ce site Web ? Par quel biais les spécifie-il dans son activité de travail quotidienne ? L'aspect lisibilité est également abordé : ses productions (prototypes fil de fer de pages et modèle StateWebCharts) seraient-elles lisibles et critiquables par le client ?

Scénario 2. Discuter des propositions d'un autre

Pour le même site Web, d'autres propositions de prototypes en fil de fer de pages et de modèle StateWebCharts ont été faites. L'une de ces propositions est présentée au sujet, qui doit les commenter à l'oral, et en extraire le meilleur, pour le combiner au meilleur de ses propres propositions. Les propositions sont très imparfaites, les prototypes de pages comportent des problèmes d'utilisabilité significatifs, le modèle StateWebCharts est faux par rapport aux pages proposées. Notamment, une même page est représentée par trois états correspondant à des opérations menées sur la page et modifiant son contenu (p.ex. confirmation de l'ajout ou de la suppression d'un enfant).

L'objectif de ce scénario est d'évaluer l'hypothèse 2 (cf. §5.3.2) qui concerne le support au consensus qu'offrent le StyleGuide et un formalisme de modélisation de la structure hypertexte. La discussion avec l'utilisateur est orientée sur les avantages et les inconvénients des choix de conception des propositions qui lui sont montrées. Cette discussion porte sur les fonctionnalités mais aussi sur des choix d'interface et l'utilisabilité. Puis l'entretien est orienté sur d'éventuelles difficultés dans le déchiffrage du modèle StateWebCharts, et l'attention du sujet est portée sur l'erreur de modélisation indiquée plus haut (une page représentée par trois états). Il est demandé si cette erreur est choquante, ou si au contraire, elle permet d'exprimer une spécificité de l'interface qui n'apparaîtrait pas si le sujet étalon avait respecté le formalisme.

Scénario 3. Proposer un ajout de* guideline *au StyleGuide

Le site Web de la cantine doit maintenant contenir une nouvelle fonctionnalité : autoriser les parents à payer les repas par carte bancaire, directement sur le site. Il se trouve que le *guideline* permettant l'insertion d'un tel formulaire n'apparaît pas dans le StyleGuide v1.7. Mais, dans un cas réel, que ce module apparaisse dans le StyleGuide ou pas, l'analyste doit l'intégrer aux prototypes de page de son application. Pour définir la page contenant le module de paiement, l'analyste va donc devoir se renseigner, faire appel à son expérience, ou à celle de ses collègues, rechercher des conseils de conception sur le Web, ou dans des livres. La réalisation de ce *wireframe* va lui coûter de nombreux efforts et lui apporter une certaine expertise sur le sujet qu'il serait bon de capitaliser et de partager avec ses collègues. Pour ce faire, il est proposé ici d'envoyer à l'équipe de gestion du StyleGuide une suggestion d'ajout de *guideline*. Il s'agit simplement d'une suggestion d'ajout, qui sera affiné et formalisé par l'équipe de gestion du StyleGuide. Cela est fait par le biais d'un canevas de *guideline* à remplir sur papier, en totalité ou partiellement.

- **Titre du *guideline* :** l'intitulé du *guideline,* tel qu'il figurerait dans le StyleGuide ;
- **Recommandations :** des conseils pratiques de mise en page, de formulation, fonctionnels, techniques, graphiques ou éditoriaux. Ces recommandations viennent en plus du prototype proposé plus bas, pour affiner ce qui y figure et éventuellement insister sur des aspects importants ;
- **Rationnel :** une justification théorique des recommandations, issue des grandes valeurs de la conception d'interface (p.ex. confort de lecture) et du bon sens de l'analyste. L'équipe de gestion du StyleGuide formalisera cette partie avec des concepts issus de leur expertise en facteurs humains ;
- **Cas d'utilisation : les** cas concrets dans lesquels ce *guideline*…
  - **doit** absolument être utilisé ;

  - **devrait** être utilisé, mais sans obligation ;
  - **ne devrait pas** être utilisé, ou plutôt, il vaudrait mieux qu'une autre alternative soit choisie ;
  - **ne doit pas** être utilisée ;
- **Exemples :** des exemples précis d'applications Web implémentant ce *guideline*
  - **positifs** : des exemples à suivre, dont il faut s'inspirer ;
  - **négatifs** : des exemples de mauvaise implémentation, où les recommandations ne sont pas suivies, où le rationnel n'est pas respecté ;
- **Ressources externes :** des références précises vers des ressources ayant nourri la réflexion de l'analyste durant le projet où ce *guideline* a été implémenté ;
- **Prototype :** un *wireframe* générique pour l'utilisation de ce *guideline*, la forme sous laquelle il pourrait apparaître dans Visio.

Ce scénario permet de valider l'hypothèse 3 (cf. §5.3.3) qui concerne la possibilité pour un utilisateur du StyleGuide d'en suggérer l'évolution. Au cours de ce scénario, il est demandé au sujet comment, en situation réelle d'utilisation du StyleGuide, il se rend compte de l'absence d'un *guideline* dont il a besoin : quels indices lui font réaliser que toute recherche supplémentaire sera vaine ? Il est demandé ensuite au sujet d'évaluer la facilité avec laquelle il a accompli cette proposition d'ajout de *guideline*, la satisfaction et la fierté qu'il en a tirées. Puis, il est demandé de réfléchir, en conditions réelles, de l'acceptabilité d'une telle activité, et éventuellement des modalités qui rendraient ce scénario plus acceptable auprès des collègues du sujet et de ses clients.

5.5. Résultats

Les résultats de cette évaluation sont présentés ci-après, avec d'abord des remarques formulées sur les données recueillies dans le pré questionnaire. Viennent ensuite des observations faites sur les scénarios réalisés par des sujets évalués, tels qu'ils ont été décrits dans la section 5.4.4. L'observation des attitudes et des activités des sujets est nourrie et corroborée par les commentaires faits à voix haute par le sujet, ainsi que les entretiens entremêlés à l'exécution des scénarios.

5.5.1. Pré questionnaires

Les données recueillies lors de ce pré questionnaire étant majoritairement quantitatives, nous avons choisi de les résumer dans un tableau, en excluant les remarques indiquées par certains sujets. Ces résultats quantitatifs sont présentés en Annexe E, cette section émet des observations à leur sujet.

Peu de recul sur l'utilisation du StyleGuide

Suite à l'analyse des pré questionnaires, nous constatons tout d'abord le peu d'expérience et de recul des utilisateurs du StyleGuide vis-à-vis de cet outil au moment de l'évaluation (mars – avril 2008). En moyenne, un sujet a participé à 0,8 projet incluant l'utilisation du StyleGuide (4 des 11 sujets n'ont participé à aucun projet – 6 ont participé à un projet – 1 a participé à deux ou trois projets) et il a personnalisé 0,9 fois un *wireframe* issu d'un guideline du StyleGuide. De même, les facilitations d'utilisation sont encore en cours de mise en place et seuls 3 sur 11 sujets ont reçu la formation à l'utilisation du StyleGuide et des formes Visio associées. Nous avons ainsi affaire à une population d'analystes (le seul sujet non analyste ayant une expérience d'analyste de plusieurs années) en poste chez SmalS depuis 21 mois en moyenne, mais avec peu de recul sur le StyleGuide.

Le caractère novice des sujets engage à la plus grande prudence quant à l'interprétation des résultats de l'évaluation. L'évaluation portera en quelque sorte sur leurs premières impressions de la méthode. Cela ne manque pas d'intérêt, notamment en ce qui concerne l'évaluation de la facilité d'apprentissage de la méthode, mais cela implique un sérieux biais dans l'évaluation de sa réelle appropriation. C'est la raison pour laquelle nous ne proposerons qu'un résumé approximatif des variables d'évaluation (cf. Tableau 8), mais accorderons une grande place à la restitution des entretiens. Des statistiques numériques seraient peu pertinentes, compte-tenu du faible nombre de participants. En revanche, les extraits d'entretiens conservent tout leur intérêt, même avec peu de sujets (Nielsen et Landauer 1993), pour détecter la majorité des problèmes d'utilisabilité.

Compatibilité et utilité ressenties comme bonnes

Tous les sujets ayant répondu au questionnaire pensent qu'ils ont les compétences nécessaires pour utiliser le StyleGuide. Tous savent à qui demander conseil pour d'éventuels problèmes d'usage, bien qu'ils n'aient pas reçu de formation. 9 des 11 sujets pensent que le StyleGuide s'intègre bien à leurs habitudes de travail. Un sujet précise :

> « La principale difficulté est de connaître les différentes manières de naviguer dans le StyleGuide. Cet apprentissage demande un certain temps. Ensuite lorsque l'on sait où trouver l'info recherchée, son utilisation est beaucoup plus simple. »
>
> -- Questionnaire 10, le 23 avril 2008.

L'un des 2 sujets qui pensent le contraire précise qu'il « manque de connaissances Visio » (Source : Questionnaire 8, le 22 avril 2008). En moyenne, les sujets sont plutôt d'accord avec le fait que le StyleGuide améliore leur travail (10 des 11 sujets sont tout à fait ou plutôt d'accord), mais sont partagés sur le fait qu'il accélère leur travail (6 des 11 sujets ne sont pas du tout ou plutôt pas d'accord). L'utilité du StyleGuide est ressentie par 10 des 11 sujets, et autant trouvent que les conseils de conception qui y sont prodigués sont compréhensibles. 8 des 11 sujets pensent qu'il est plutôt facile, au sein du StyleGuide, de trouver un *guideline* adéquat.

Appropriation mitigée

L'appropriation du StyleGuide par ses utilisateurs est plus mitigée : 6 des 11 sujets n'utiliseraient pas le StyleGuide s'ils avaient une autre alternative. Le gain de considération pour un utilisateur du StyleGuide n'est pas approuvé non plus. Cependant, en moyenne, un sujet a utilisé 1,2 fois le StyleGuide comme support à son argumentation, en phase de conception (5 des 11 sujets ne l'ayant toutefois jamais fait). Ce recours au StyleGuide comme support argumentatif, chez des sujets pourtant novices dans son utilisation, représente un signe positif de son acceptabilité.

5.5.2. Utilisation du StyleGuide et des formes Visio associées

Le premier scénario demandé aux 8 sujets qui ont été observés consistait en la réalisation d'un *wireframe* de page, pour une application fictive donnée (cf. §5.4.4), avec le support du StyleGuide et des formes Visio. La version du StyleGuide proposée lors de l'évaluation est la version 1.7 ; les sujets utilisent des versions différentes les uns des autres, mais ces versions ont la même forme que la version 1.7 présentée. Les *guidelines* et les formes Visio sont différents mais on retrouve les mêmes catégories générales dans le StyleGuide.

Difficultés dans l'utilisation du StyleGuide

Utilisation du StyleGuide

Le StyleGuide a été peu utilisé par l'ensemble des sujets, au contraire des formes. Seulement 4 des 8 sujets ont consulté au moins une page du StyleGuide pour réaliser le premier scé-

nario. L'évaluateur présentait pourtant les consignes de la façon suivante : réalisation d'un *wireframe* de page avec le support du StyleGuide (l'évaluateur montre le StyleGuide ouvert dans un navigateur Web) et des formes Visio associées (l'évaluateur montre les formes dans Visio). Le poste était ensuite laissé à l'utilisation du sujet, la fenêtre du StyleGuide en premier plan. Les enregistrements d'écran montrent ainsi que, même pour les sujets ayant consulté le StyleGuide lors de l'évaluation, cette utilisation reste anecdotique (6 des 8 sujets ayant consulté moins de 4 pages du StyleGuide). Le Tableau 7 présente le nombre d'interactions relevées avec le StyleGuide lors de la réalisation du scénario n°1 de réalisation d'un *wireframe* de page.

- Chaque ligne correspond à un sujet observé, identifié dans la 1ère colonne ;
- La 2ème colonne fait apparaître le nombre de clics (activation de lien hypertexte) relevés pour la navigation dans le menu ; ces clics provoquent le déploiement d'un groupement de *guidelines* ou le passage d'une vue à l'autre ;
- Le nombre de pages affichées (*guideline* ou page auxiliaire, comme une table des matières ou une page d'aide) apparaît dans la 3ème colonne ;
- La 4ème et dernière colonne montre le nombre de pages lues : une page est dite « lue » lorsqu'elle n'a pas été simplement affichée, mais consultée par le sujet.

Tableau 7. Interactions avec le StyleGuide relevées lors de la réalisation du scénario n°1

Sujet	Navigation dans le menu (nb. de clics)	Pages affichées	**Pages lues**
n°1	15	14	**13**
n°2	0	0	**0**
n°3	1	2	**2**
n°4	1	0	**0**
n°5	13	6	**4**
n°6	0	0	**0**
n°7	0	0	**0**
n°8	13	3	**2**

Navigation selon différentes vues

Le StyleGuide propose différentes vues, une pour chacun des métiers susceptibles de l'utiliser (analyste, graphiste, responsable éditorial, développeur). Ces vues ont du sens pour les utilisateurs analystes : ils doivent considérer les autres points de vue dans leur spécification fonctionnelle, pour proposer des solutions exploitables par les autres composantes de la conception.

> « Donc, moi je vais surtout voir le F et le G [NDLR : vues analyste fonctionnel et graphiste]. Le T [NDLR : vue développeur technique], ça je regarde presque jamais. »
> -- Sujet 1, le 15 avril 2008.

> « Je trouve bien que [le StyleGuide] soit séparé par différents types de fonctions, comme ça tu peux aller directement vers ce dont tu as besoin dans les différentes étapes. Là, par exemple, quand j'ai commencé à faire les *wireframes*, ben c'était plutôt le fonctionnel que je regardais, et maintenant, je suis plutôt en train de regarder pour graphiste et pour développeur. »
> -- Sujet 2, le 16 avril 2008.

Le sujet 1, comme le sujet 2, utilise le StyleGuide principalement selon la vue « Analyste fonctionnel » qui correspond à son statut, mais pas uniquement. Le sujet 2 en particulier précise que, selon le moment de la conception, il consulte le StyleGuide suivant d'autres vues, par exemple la vue « Développeur » lorsque le maquettage interactif approche. En plus de tirer parti des catégories définies dans la colonne de menu (cf. Figure 51: *Define application purpose*, puis

Choose interface type, etc.), ce sujet hiérarchise les vues en suivant le même type de catégorisation : l'ordre naturel de conception.

Recherche de *guideline*

Les sujets éprouvent cependant des difficultés à localiser les *guidelines* contenus dans le StyleGuide. Ceux qui ont participé à au moins un projet concret avec le StyleGuide ont l'impression d'en avoir une vision globale. Certains d'entre eux ont commencé par lire le StyleGuide dans son intégralité lors de leur premier projet. C'est ainsi que, quand ils sont à la recherche d'un *guideline* particulier, ils se servent avant tout de leur connaissance du StyleGuide, pour savoir si le *guideline* figure dans le StyleGuide. L'organisation du StyleGuide et les intitulés des *guidelines* ne les aident pas particulièrement dans cette tâche. D'ailleurs, lorsque leur recherche n'aboutit pas, ils recourent à un balayage exhaustif et linéaire des *guidelines*. Ils ont également le réflexe de contacter les membres du centre de compétence en utilisabilité pour demander leur aide.

> « – Là, je me perds un peu… (…)
> – Qu'est-ce qui te fait penser, parfois, qu'un élément que tu cherches n'est pas dans le StyleGuide ?
> – Peut-être qu'il y est mais que je ne sais pas où aller le chercher ! (…) Tu peux même avoir un truc pour aller chercher… Quand tu sais pas très bien le bon nom… C'est déjà parfois un problème… "Ça s'appelle comment ?!" (…) Alors je vais chercher tout et je vais regarder partout pour voir… "Est-ce que c'est là-dedans ?..." »
> -- Sujet 1, le 15 avril 2008.

> « [Une chose à améliorer dans le StyleGuide], c'est la navigabilité. Faut dire qu'on ne trouve pas toujours les choses quand on veut les trouver ! Ce que j'ai fait d'abord – parce que je suis débutant, donc je ne savais rien du tout – j'ai pris tout le StyleGuide et je l'ai lu en entier. Donc, je savais plus ou moins tout le contenu. Mais après j'ai essayé de chercher ce que je voulais et… J'y ai passé un bon moment… Finalement, [un membre de l'équipe *Usability*] m'a justement appelé pour cette évaluation, et je lui ai posé la question. (…) Aussi, les noms ne sont pas très significatifs… Par exemple, je savais qu'il y avait un truc avec les erreurs, ben c'est "*input error*" donc moi je mettrais plutôt "erreur" d'abord. »
> -- Sujet 2, le 16 avril 2008.

> « Et là, [l'utilisateur] aura un calendrier avec les dates pour lesquelles il y a des repas réservés pour cet enfant-là. (…) Donc j'irais voir ici ce qui est en rapport avec les calendriers. Par contre, je ne sais pas où c'est, donc c'est là que les ennuis commencent ! (Rires) (…) J'ai une connaissance globale du StyleGuide, ce qui fait que je me doute de ce qu'il y a dedans et de ce qu'il n'y a pas... Puis, si vraiment je ne trouve pas, je téléphone au *Usability Competency Center*. »
> -- Sujet 4, le 17 avril 2008.

> « [Ce qui me ferait penser qu'un *guideline* dont j'ai besoin n'est pas dans le StyleGuide ?] Eh bien, je ne sais pas… Le fait que je suis déjà allé sur le site et que je n'ai jamais vu ça. »
> -- Sujet 6, le 17 avril 2008.

Les sujets 2, 4 et 6 évoquent leur connaissance du StyleGuide comme arme principale pour savoir si un *guideline* y apparaît ou pas. Les sujets 2 et 4 complètent cet *a priori* par un recours direct au centre de compétences en utilisabilité. Les sujets 1 et 2 évoquent encore un balayage exhaustif du StyleGuide, technique systématique qui sera rendue de plus en plus laborieuse au fur et à mesure que le volume du StyleGuide grandira… Une question de terminologie est soulevée par les sujets 1 et 2, et peut être distinguée au travers des réflexions du sujet 4 qui cherche un « calendrier », et qui pourrait trouver des informations sur un *guideline* qui traite de « date ». Un thésaurus est en cours de mise en place chez SmalS, le coupler aux moyens de navigation serait un atout pour compenser ces problèmes de terminologie.

Utilisation des formes Visio efficace mais non optimale

Appropriation de Visio

L'appropriation de l'outil Visio n'est pas évidente. Bien que l'évaluateur invite les sujets, en premier lieu, à reconfigurer à leur goût l'outil et la disposition des formes (cf. §5.4.4), ils ne personnalisent pas ou personnalisent légèrement l'outil. La personnalisation est celle proposée en formation, ou vise l'allégement des informations disponibles à l'écran.

> « J'affiche dans Visio les boîtes dont j'ai besoin, et j'ajoute au fur et à mesure ce dont j'ai besoin. (…) Une fois que tu as fait ton *design page* par exemple, tu n'as plus besoin de celle là. »
> -- Sujet 1, le 15 avril 2008.

> « Alors, oui, moi je préfère la visualisation avec les icônes… »
> -- Sujet 2, le 16 avril 2008.

Les sujets 1 et 2 évoquent des préférences dans l'utilisation des formes du StyleGuide. Il s'agit de les afficher de façon plus facile à explorer, soit en en affichant des aperçus, soit en réduisant le nombre. Il ne s'agit pas d'une personnalisation pour plus d'efficacité ou de densité, comportement que l'on aurait pu attendre de la part d'utilisateurs expérimentés.

Utilisation de Visio

Des difficultés subsistent dans l'utilisation du logiciel Visio lui-même, difficultés qui seront probablement levées lorsque les sujets auront participé aux formations au StyleGuide et à l'utilisation de ces formes. La pratique de l'outil aidera également à l'apprentissage de son usage, car jusqu'à il y a quelques mois, la norme dans l'entreprise était de travailler avec un autre logiciel d'édition de schéma, nommé iGrafx[1]. Certains sujets évoquent ainsi des habitudes de travail à perdre pour se servir correctement de Visio. À ces difficultés, s'ajoutent des problèmes de version, avec des terminologies qui changent d'une version à l'autre, et des intitulés qui ne renvoient plus la forme attendue. Des frustrations découlent de ces défauts d'utilisabilité de Visio.

> « Je vais voir lequel je mets, j'ai l'impression que ce n'est pas tout à fait le même. (…) Ça c'est un problème : tu dois à chaque fois aller voir lequel c'est !! Tu prends un *datagrid* et tu ne sais pas bien lequel c'est. (…) Tous les deux s'appellent *datagrid* mais c'est quoi la différence entre les deux ? (…) Je vais les mettre l'un sur l'autre… Ça ne va pas être *bien*… Pour voir duquel j'ai besoin, lequel je peux utiliser… »
> -- Sujet 1, le 15 avril 2008.

> « Est-ce qu'il y a alors des choses plus simples ou pas ? Non, y'a pas vraiment… Bon, ben, tant pis, je vais utiliser le gros *datagrid* alors (…) Ah… Avant il diminuait ou il augmentait le nombre de colonnes selon qu'on le rétrécissait ou qu'on l'agrandissait. »
> -- Sujet 8, le 15 avril 2008.

> « J'aimerais bien éditer *row1*, *row2*, *row3*, mais je ne sais plus, ou je n'ai jamais su… Comment je peux modifier ? (…) Bon, ben, par contre, je ne pourrai pas mettre de *tabs* et ça c'est dommage… (…) Pour l'instant, on va laisser comme ça, sinon on ne va pas avancer. »
> -- Sujet 3, le 16 avril 2008.

> « Ah, il l'a pas mis au bon endroit… Tu vois, il faut éditer ce truc-là… En fait, oui, c'est ça qui est compliqué… (…) Quand tu as des *shapes* qui sont regroupées, c'est très très compliqué pour aller éditer le bon bazar… Il y a de quoi s'énerver beaucoup ! »
> -- Sujet 4, le 17 avril 2008.

[1] http://www.igrafx.fr/ (dernier accès : 25 sept. 08)

Les sujets 1 et 8 évoquent des formes trop complexes relativement à leurs besoins. Le problème de la difficulté de l'édition de formes groupées revient également chez plusieurs sujets, et notamment chez les sujets 3 et 4 cités ci-dessus. Il s'agit d'un problème d'utilisabilité du logiciel Visio, mais qui ne doivent pas être négligés dans la mesure où le logiciel est un support incontournable à l'utilisation des formes du StyleGuide.

Sobriété des formes proposées

Certains précisent que les formes Visio permettent une anticipation sur l'apparence finale de l'application. La sobriété graphique des formes proposées ne gêne pas la majorité des sujets, bien au contraire : leur rôle est de définir l'interface fonctionnelle et ils apprécient de ne rien parasiter de cela lorsqu'ils sont face au client.

> « Dans la plupart des cas, moi en tous cas, on commence avec ce qui doit être sur l'écran, donc on commence à faire une proposition sans tous les trucs [NDLR : *header, footer*] autour. Sinon [les clients] commencent : "Ah c'est pas comme ça, c'est pas la bonne couleur..." »
> -- Sujet 1, le 15 avril 2008.

> « Je mets tous les composants en place, et puis je me dis… Bon, visuellement, c'est pas très joli, qu'est-ce que je vais faire ? (…) J'ai au moins une base. Si je devais aller revoir le client, je peux déjà discuter avec lui parce qu'il voit ce dont il avait besoin. Après, il va me dire "C'est pas très joli…" OK, on en rediscutera. Par contre, si je lui mets juste le *header* avec de belles couleurs, il va me dire "Et mon application, j'en fais quoi ?" »
> -- Sujet 3, le 16 avril 2008.

> « Je viens encore d'avoir une discussion avec l'un de mes analystes qui a suivi la formation StyleGuide à ce sujet. Il semblait déçu que finalement les analystes ne puissent plus toucher au graphisme des écrans et il avait l'impression qu'alors il allait proposer au client des prototypes pas professionnels... Personnellement, je ne suis pas du tout d'accord avec ça : rien de moins professionnel que du graphisme fait avec des outils non professionnels (Paint) par des non professionnels du graphisme ! Et en plus, cela empêche le client qui doit faire part de son *feedback* sur les prototypes de se concentrer sur le fonctionnel. »
> -- Sujet 4, courrier électronique du 22 avril 2008.

> « Des *tabs* ou des *radio buttons*… C'est un parti pris de *designer*, ça ne relève pas de l'analyse. Pour moi, c'est une question purement de présentation. Il faut savoir qu'on doit choisir : ça c'est de l'analyse. »
> -- Sujet 5, le 18 avril 2008.

Les sujets 1, 3, 4 et 5 s'accordent sur la responsabilité de l'analyste : définir la spécification fonctionnelle et non graphique. Cependant, le sujet 4 rapporte un avis contraire dans un courrier électronique consécutif à l'expérimentation. Des analystes, en effet, se voient frustrés d'être privés de cet aspect graphique. La communauté d'analystes de l'entreprise n'est donc pas convaincue à l'unanimité de la nécessaire sobriété graphique d'un *wireframe*, telle que le suggèrent les formes du StyleGuide.

Adhésion aux principes de standardisation et de réutilisation

Standardisation

La standardisation des interfaces produites au fil des projets fait sens dans l'esprit des analystes. Ils apprécient qu'une ligne directrice ferme soit donnée pour la conception de ces interfaces. Ils l'apprécient parfois pour des raisons d'utilisabilité mais surtout pour établir une cohérence interne à l'entreprise : ils se trouvent guidés vers des solutions qu'il savent approuvées par leur direction.

> « [Un avantage du StyleGuide] c'est que [les *wireframes* que tu produis] doivent être validées par l'*usability*, donc tu as accès à ce qui est la bonne pratique des choses. Donc après pour la validation, ça va mieux. (…) [Ce que je trouve bien c'est que le StyleGuide] met un standard. Chacun ne fait pas les choses à sa façon, mais il y a quelque chose qu'on suit, et tout le monde, plus ou moins, fait ce qui lui plaît sur cette base. »
> -- Sujet 2, le 16 avril 2008.

> « Le StyleGuide améliore la cohérence au niveau des applications qu'on fait. Puisqu'on travaille toujours plus ou moins pour les mêmes clients, donc… »
> -- Sujet 4, le 17 avril 2008.

Le sujet 2 apprécie de savoir par avance, de façon formalisée et illustrée ce que préconise le centre de compétences en utilisabilité. Ce même sujet évoque la standardisation comme avantage du StyleGuide : il parle de façon positive d'un cadre fortement guidant pour tous les concepteurs. Le sujet 4 fait preuve d'un grand recul vis-à-vis de l'utilisabilité des applications d'e-Gouvernement avec une allusion à la cohérence entre interfaces d'une application à l'autre.

Guidage

Les sujets relèvent un guidage fort apprécié dans les toutes premières étapes de la réalisation de leurs *wireframes* : le StyleGuide leur donne alors des réponses très concrètes, en particulier dans le choix des éléments d'interface à utiliser, ou dans la disposition de ces éléments dans la page.

> « [Le principal avantage du StyleGuide, c'est le support pour la conception d'interfaces, par exemple] parfois, comment mettre les choses sur l'écran… Quelles est la meilleure façon de mettre tel ou tel truc sur l'écran. »
> -- Sujet 1, le 15 avril 2008.

> « [Le principal avantage du StyleGuide], c'est que ça donne des règles de base très claires sur… Enfin, ça permet de se débrouiller pour démarrer. (…) Ça ne va pas régler tous tes problèmes : après, quand les écrans se complexifient… Tu es dans un contexte particulier… (…) Mais au moins avec le StyleGuide, tu te dis : "Est-ce que c'est une *dropdown list* ou est-ce que c'est, je ne sais pas, un *checkbutton* ?". Eh bien, là au moins, tu as des réponses claires sur ce genre de trucs… Mais assez basiques. Mais c'est suffisant je pense. Tu ne saurais pas formaliser au-delà. »
> -- Sujet 4, le 17 avril 2008.

> « [Le principal avantage du StyleGuide c'est que] ça permet de guider pour la réalisation des écrans. »
> -- Sujet 5, le 18 avril 2008.

Les sujets 1, 4 et 5 apprécient d'être guidés dans les premières heures de la conception, lorsqu'il s'agit de placer les premiers éléments de l'interface. Le StyleGuide et les formes Visio associées leur offrent une base, leur permettent de ne pas partir de rien. Le sujet 1 évoque l'aide pour la disposition des éléments de l'interface, tandis que le sujet 4 insiste sur le guidage pour le choix des éléments d'interface eux-mêmes, relativement à leur fonction.

Flexibilité

La flexibilité fournie par les formes Visio est appréciée par les analystes : ils pensent que c'est un outil efficace et rapide. En laissant de côté les difficultés de manipulation de l'outil lui-même dont nous parlerons plus loin, Visio et les formes du StyleGuide permettent de produire facilement et rapidement une première proposition de *wireframe*. De plus, l'outil permet de prendre en compte rapidement et sans grande complexité les modifications demandées par le client.

> « Notre client, on est allé le voir plusieurs fois, et plusieurs fois on a dû faire des modifications. Dans mon équipe, avant l'apparition [du StyleGuide], ils fonctionnaient avec du XHTML directement. Ce qui veut dire que, quand on avait une modification à faire, c'était vraiment pas évident parce que c'est du code XHTML, c'est pas du *drag and drop* et du clic gauche – clic droit. (…) Alors que nous, avec Visio, on a pu refaçonner complètement ce que voulait l'utilisateur. On a fait une première proposition, ils ont voulu des changements, on a dû faire une deuxième proposition, et on aurait pu en faire une troisième, une quatrième très rapidement. »
> -- Sujet 3, le 16 avril 2008.

> « Ce qui est bien aussi, évidemment, c'est ce qu'ils ont fait dans le StyleGuide en développant des *templates*, des modèles de maquettes, ça c'est un gain de temps et de productivité. Et ça facilite le travail ! »
> -- Sujet 4, le 17 avril 2008.

Le sujet 3, un participant aux projets pilote du StyleGuide, a grandement apprécié la flexibilité offerte par les formes Visio pour la conception de *wireframes* et surtout pour les modifications qui ont dû y être successivement apportées. Le sujet 4 le rejoint sur les questions de productivité.

Intégration des principes d'utilisabilité

Place du centre de compétences en utilisabilité

Les sujets évoquent un recours systématique au centre de compétence en utilisabilité pour l'évaluation de l'utilisabilité de leurs propositions et des corrections éventuelles. Conscients que leurs *wireframes* passeront cette étape de validation, ils ne cherchent pas à trancher certaines décisions qui leur semblent relever de la compétence de ce service.

> « [La disposition du bouton Valider], c'est à discuter avec l'*usability*, généralement. On leur soumet des maquettes de ce genre et on leur demande ce qui se fait au niveau *best practices* et tout ça. Par exemple, ici, il mettra un bouton lorsqu'une action suppose une introduction dans une base de données, ce genre de choses. »
> -- Sujet 7, le 15 avril 2008.

> « On n'a pas toujours le temps d'aller tout re-vérifier dans le StyleGuide, donc parfois c'est pour ça qu'on repasse par l'équipe de *usability* pour être certains que ce qu'on fait correspond bien au StyleGuide. »
> -- Sujet 8, le 15 avril 2008.

Les sujets 7 et 8 conçoivent leur *wireframe* sans chercher à trancher eux-mêmes sur certains points pour lesquels ils pressentent que l'utilisabilité ne sera pas optimale. Le centre de compétences en utilisabilité inspectera le *wireframe* et proposera très probablement des corrections. Quoiqu'ils discernent les éléments qui rendent leur *wireframe* peu utilisable, ils n'essayent pas de les corriger eux-mêmes, parce qu'ils manquent de temps et parce qu'ils ont toute confiance en l'expertise du centre de compétences en utilisabilité. Le sujet 8 évoque pourtant le fait que les informations de bonne pratique sont à sa disposition dans le StyleGuide.

Culture en utilisabilité

La culture en utilisabilité des sujets observés est bonne. Bien qu'ils ne soient pas experts en utilisabilité, leurs critiques des propositions étalon sont pertinentes et argumentées. Les sujets se servent d'ailleurs des corrections réalisées sur leurs *wireframes* dans des projets précédents, des remarques que le centre de compétence en utilisabilité a formulées sur des propositions précédentes. Certains sujets relèvent que ces conseils de conception corroborent le contenu du StyleGuide : ils ont donc potentiellement, via le StyleGuide, la compétence de l'équipe *usability* à leur disposition.

« Je me base surtout sur d'autres écrans, sur des choses que j'ai déjà vues et qui, à moi, me semblent faciles. Et après, il y a l'*usability* qui dit "Non." (Rires) (…) Je viens juste d'avoir des remarques sur des écrans que j'ai faits… Donc pour l'instant c'est "ah, ça je ne peux plus faire, ça je ne peux plus faire…" »
-- Sujet 1, le 15 avril 2008.

« [J'ai noté des différences dans les projets], pas nécessairement avec le StyleGuide mais dans le fait de se faire accompagner par l'équipe *usability* qui, de toutes façons, elle, se base notamment sur le StyleGuide. (…) Ils ont vraiment apporté des choses intéressantes, concrètes, des améliorations qui parfois ne coûtent pas cher… Et il y a des choses qu'on ne voit pas ! Parce que quand on a le nez dans ses écrans, il y a des trucs qu'on ne voit absolument pas. (…) On avait fait aussi des *usability tests* et c'est vrai que… ça nous met le nez dessus ! Quand on voit l'utilisateur qui dit "Ah non… Je ne vois absolument pas ce que veut dire cet icône…", on se dit "Mais, enfin ! C'était pourtant évident !!". Eh bien non ! (Rires) »
-- Sujet 4, le 17 avril 2008.

« Ici, j'aurais pu mettre aussi un bouton, mais je ne l'ai pas mis parce qu'on nous avait dit qu'un bouton, c'était vraiment quand on validait quelque chose, qu'il y avait un envoi, etc. Et ça, c'est dans le StyleGuide. »
-- Sujet 8, le 15 avril 2008.

Le recours systématique à l'expertise du centre de compétences en utilisabilité que nous avons évoqué plus haut, n'empêche pas les concepteurs de tirer des leçons des corrections qui leur sont proposées et de les intégrer à leurs travaux suivants. Ainsi, les sujets 1 et 8 évoquent des reproches qui leur ont été faits sur des projets précédents et en tiennent compte pour le *wireframe* qu'ils conçoivent lors de l'expérimentation. Le sujet 4 constate les bienfaits d'un œil extérieur pour l'évaluation de l'utilisabilité de ses *wireframes*. Des tests utilisateurs lui ont aussi offert un recul sur sa production, et ont permis d'appliquer des corrections auxquels il adhère à sa grande surprise, parce qu'il n'aurait pas pu les établir seul.

5.5.3. Perception du StyleGuide

Défauts de communication

Les sujets, dans leur grande majorité, ne connaissent pas les évolutions du StyleGuide, et la sortie des dernières versions. Ces informations leur sont données en formation, mais, ensuite, ils ne sont pas tenus au courant des mises à jour du StyleGuide et de ses formes. Il y a notamment une frustration quant au fait que leur formation au StyleGuide vienne très tard, souvent bien après qu'ils aient commencé à utiliser le StyleGuide dans leur activité quotidienne.

« *Icons and images*, ah tiens, tu as ça, moi je ne l'ai pas… Je n'ai pas encore trouvé mes icones. (…) Chez moi, il n'était pas dans [l'archive contenant toutes les formes Visio associées au StyleGuide], peut-être que je dois retourner voir encore une fois s'il y a une nouvelle archive… »
-- Sujet 1, le 15 avril 2008.

« Observation générale sur le StyleGuide, eh bien peut-être prévenir de son évolution ! Parce que, moi je suis sur [une ancienne version] encore ! (...) Je n'ai pas reçu, à ma connaissance, d'email me disant "Attention, il y a un nouveau site". »
-- Sujet 4, le 16 avril 2008.

« Et ça c'est la version du StyleGuide qui est à la SmalS ? Ah bon ! Je suis heureux de le savoir !! (Rires) Tu vas m'envoyer le lien ? »
-- Sujet 5, le 18 avril 2008.

Les sujets 4 et 5 n'étaient pas au courant des dernières versions du StyleGuide publiées. C'est d'autant plus fâcheux que le sujet 5 avait, par ailleurs, évoqué des problèmes de *bugs* qui ont été corrigés dans les versions suivant celle qu'il connaît. Le sujet 1, quant à lui, parle de la

nécessité d'aller voir régulièrement s'il n'y a pas de nouvelles versions du StyleGuide. Il emploie l'expression « retourner voir *encore une fois* » qui laisse à penser que cela l'ennuie.

Engagement vis-à-vis du StyleGuide

Responsabilité dans l'évolution du contenu

Les sujets sont partagés quant à leur responsabilité vis-à-vis du devenir du StyleGuide. Beaucoup de sujets sont convaincus du fait que le StyleGuide est l'affaire du centre de compétence en utilisabilité. Ils reconnaissent avoir leur place en tant que participant mais ne se sentent pas responsable de l'évolution de son contenu. Ils constatent également que l'évolutivité du StyleGuide est l'un de ses atouts majeurs, et que, par nature, il se doit d'être flexible et d'évoluer au fil du temps.

> « [Ma réponse à "Je me sens responsable de l'évolution du StyleGuide"] n'est pas ["Pas du tout d'accord"] parce que, ce que je vais proposer, je vais le tenir à l'œil mais le reste… Si ce n'est pas moi qui m'en occupe, forcément, je ne vais pas m'en sentir responsable… Ce n'est pas pour autant que je ne me sentirai pas intéressé."
> -- Sujet 3, le 16 avril 2008.

> « [Je me sens responsable de l'évolution du contenu du StyleGuide :] si le StyleGuide n'évolue pas, il ne sera plus utilisable. »
> -- Sujet 4, le 17 avril 2008.

> « [Suggérer un ajout de *guideline* m'a semblé plutôt gratifiant :] si on y passe du temps, autant que ça serve aux autres. »
> -- Sujet 6, le 17 avril 2008.

> « [J'ai trouvé gratifiant de suggérer un ajout de *guideline*,] c'est sympa de pouvoir aider ! (…) [Je me sens responsable de l'évolution du contenu du StyleGuide] parce qu'on est en première ligne, dans un sens. »
> -- Sujet 7, le 15 avril 2008.

> « [Faire évoluer le contenu du StyleGuide,] c'est pas mon boulot à la base (sourire), en tous cas on a aussi d'autres choses à faire. (…) Le fait qu'on puisse proposer des choses ça me parait logique, parce qu'on est parfois confronté à des situations particulières dans les projets, mais je ne me sens pas responsable [du StyleGuide] pour autant. »
> -- Sujet 8, le 15 avril 2008.

Tous les sujets interrogés pensent que leur expérience et leurs connaissances servent à leurs collègues et ont, de ce fait, trouvé gratifiant de proposer d'ajouter un *guideline* au StyleGuide, comme le disent les sujets 6 et 7. Les sujets 4 et 7 s'entendent sur le fait que le StyleGuide ne vit que s'il évolue, et le sujet 8 les rejoint pour dire qu'ils sont les principaux initiateurs des évolutions. Mais quant à leur responsabilité dans l'évolution du StyleGuide, le sujet 3 se sentira responsable seulement des suggestions qu'il aura faites, et le sujet 8 évoque des freins matériels à sa participation au StyleGuide, sur lesquels nous nous attardons dans la section suivante.

Faisabilité de suggestions ponctuelles de *guidelines*

Le scénario 3 met les sujets face à un problème de conception d'interface non référencé dans le StyleGuide, pour lequel ils devront faire appel à d'autres ressources que ce dernier pour réaliser des *wireframes* pertinents. Ils doivent alors remplir un formulaire de suggestion d'ajout de *guideline* (cf. §5.4.4). Cette activité est déclarée plutôt ou tout à fait gratifiante à l'unanimité par les 9 sujets. Cependant ils doutent, en situation réelle, de l'acceptabilité d'une telle activité : leur temps de travail est fléché sur des budgets spécifiques, et ils n'ont pas l'occasion de passer du temps à ce genre d'activité transversale. La majorité des sujets le regrette, d'ailleurs, pour ce cas précis.

> « La question [à propos des suggestions d'ajout au StyleGuide,] c'est "Est-ce que ce serait accepté par tout le monde ?" »
> -- Sujet 1, le 15 avril 2008.

> « Là, ça ne m'a pas pris beaucoup de temps à [suggérer un *guideline*] parce que c'était une invention mais normalement, je pense que ça devrait être fait très détaillé, quoi. Pas une chose qu'on fait en vingt minutes, ou trente minutes, une chose pour laquelle on prend une heure, une heure et demie, deux heures s'il le faut. Le temps qu'il faut quoi, bon ça dépend de ce que tu veux ajouter. (…) Tout ça dépend des circonstances, mais je serais prêt à le faire. »
> -- Sujet 2, le 16 avril 2008.

> « [Je pourrais proposer un ajout de *guideline* dans mon activité quotidienne ?] Plutôt oui… Ça dépend du *guideline* à ajouter. S'il est très grand, je ne pourrai certainement pas parce qu'ici on fonctionne par budget / projet, donc ça veut dire que tout le temps que je dépense sur une journée, c'est sur un projet particulier, c'est un client particulier qui assume ce coût. Ce qui fait que ce client-là, il n'en a peut-être pas grand-chose à faire… (…) Ça dépend du client, du chef de projet, ça dépend de beaucoup de choses. »
> -- Sujet 4, le 16 avril 2008.

> « Je suis parfaitement conscient de l'intérêt de ça, mais généralement la *deadline* du projet prime sur le reste. »
> -- Sujet 6, le 17 avril 2008.

Les sujets 1 et 4 parlent de l'acceptabilité d'une telle activité dans leur travail quotidien et la mettent en doute, sur certains projets où les contraintes matérielles sont strictes. De plus, le sujet 2 estime que suggérer un *guideline* doit être fait avec application, et donc que cette activité doit prendre le temps nécessaire. Or le temps fait souvent défaut, comme le fait remarquer le sujet 6.

Formulaire de suggestion

Les informations demandées dans le questionnaire de suggestion de *guideline* (cf. Annexe D) sont indiquées, pour la plupart, comme facultatives. Il est encore précisé oralement au sujet qu'il ne doit pas remplir tous les champs, mais seulement ceux qui lui semblent pertinents pour son cas précis, puisqu'il s'agit d'un questionnaire très générique.

> « [Suggérer un *guideline*] était facile… Avec des explications ! (Rires) Parce que si on te donne ça et qu'on te dit "Voilà !", c'est pas évident… Il faudrait mettre un exemple. »
> -- Sujet 2, le 16 avril 2008.

> « [Suggérer un *guideline*] était facile… Oui, sauf qu'il y avait beaucoup d'informations par rapport à ce que je pouvais [renseigner]. »
> -- Sujet 4, le 17 avril 2008.

> « Il y a peut être trop qui est demandé par rapport à ce qu'on peut vraiment proposer. (…) Ça serait peut être plus simple si ça se limitait à un petit texte explicatif, avec "Voilà ce qu'on propose" et puis qu'éventuellement quelqu'un passe après pour définir un cadre formel. (…) Ce serait plus simple avec une personne de contact. »
> -- Sujet 8, le 15 avril 2008.

Le sujet 2 relève que, sans l'aide de l'évaluateur il n'aurait pas pu remplir le formulaire de suggestion. Cette remarque doit faire l'objet d'attentions, faute de quoi l'analyste abandonnera cette tâche, en particulier s'il manque de temps. Les sujets 4 et 8 soulignent que le nombre d'informations à remplir est trop important, que certaines informations les dépassent, et ce alors que l'évaluateur a bien précisé que la plupart des champs étaient facultatifs.

5.5.4. Réalisation de modèles StateWebCharts

Pertinence de la représentation de la structure hypertexte

Tous les sujets représentent la structure hypertexte de l'application Web dans leurs activités quotidiennes de conception, et s'accordent à dire que cette représentation aide à voir l'application dans sa globalité. Les analystes interrogés sont également responsables de la spécification des données de l'application. Des outils méthodologiques sont imposés par l'entreprise mais leur laissent une marge de manœuvre : plusieurs analystes reconnaissent d'ailleurs utiliser une notation « maison ».

> « On fabrique un *state diagram* [ou quelque chose d'approchant]. Pas de différence entre les [états] statiques et non statiques, mais du reste c'est un peu comme [StateWebCharts]. »
> -- Sujet 1, le 15 avril 2008.

> « Ça ressemble à un diagramme d'activités avec des annotations en plus : des pré conditions, ce genre de choses. (…) En général, je fais [une colonne] par… par exemple, si c'est l'utilisateur ou le système qui interagit, j'ai un cadre qui spécifie le *scope*, ou si il y a un système tiers, externe. »
> -- Sujet 6, le 17 avril 2008.

> « J'utilise un mélange d'UML et de ce qui me semble le plus clair. Des diagrammes de séquence. Donc, ça revient toujours au même, des carrés avec des flèches. (…) C'est surtout pour représenter la chronologie des écrans, donc le flot et ajouter des informations supplémentaires[, plus techniques]. »
> -- Sujet 7, le 15 avril 2008.

La norme dans l'entreprise est d'utiliser les diagrammes d'activités, comme le décrit le sujet 6. D'autres entretiens révèlent aussi l'utilisation de diagrammes de classe ou entité – relation pour modéliser les éléments de l'interface, en relation avec la base de données. Les sujets 1 et 7 évoquent la représentation du flot d'écrans dans leur activité quotidienne, avec des formalismes proches de StateWebCharts.

Défaut dans la représentation des données

Les analystes interrogés sont ainsi, dans leur activité quotidienne, en charge de la spécification de la base de données, et ils évoquent l'absence de spécification des données dans le formalisme StateWebCharts. Leur retour envers l'équipe technique doit être suffisamment cadrée pour que leurs besoins soient pris en compte lors du maquettage interactif puis de la conception de l'application finale. Ils précisent notamment les pré conditions sur des actions, et les appels à la base de données pour le remplissage des champs par exemple.

> « Un exemple d'écran avec des numéros, et sur chaque numéro, je mets [le type d'objet], est-ce que c'est *enabled* ou *disabled*, quelle est la source de données… En gros, hein… Parce que parfois, si le modèle de données n'est pas terminé, il y a des choses qu'on ne peut pas préciser. L'événement ou l'action, par exemple "permet à l'utilisateur d'identifier la formule dans toute l'application" (…), *default value* s'il y en a une, et les règles de validation (…) avec les messages d'erreurs[, standardisés,] si la règle n'est pas respectée. (…) Les développeurs étaient contents que ce soit bien précis. Ceci dit c'est peut-être parfois trop précis, parce que c'est difficile à maintenir, (…) puis ça ne laisse pas de liberté au développeur. »
> -- Sujet 4, le 17 avril 2008.

> « Moi ce que j'aime bien représenter là-dessus, c'est que je donne des numéros aux boutons, aux zones dans lesquelles on peut interagir, et j'aime bien les faire apparaître dans les enchaînements d'écrans. »
> -- Sujet 7, le 15 avril 2008.

« Dans le cadre de ce projet là, avec mon collègue, on avait fait un fichier un peu spécial qui s'appelait "*Requirements* techniques » qui définissait clairement, dans chaque interface, que faisait quoi. Mais ce n'est pas vraiment un standard, même chez SmalS. Le but, c'était vraiment de faire le lien entre la base de données et les écrans en termes d'enchaînement d'écrans, en termes de remplissage des champs. [NDLR : cf. Figure 55] »
-- Sujet 8, le 15 avril 2008.

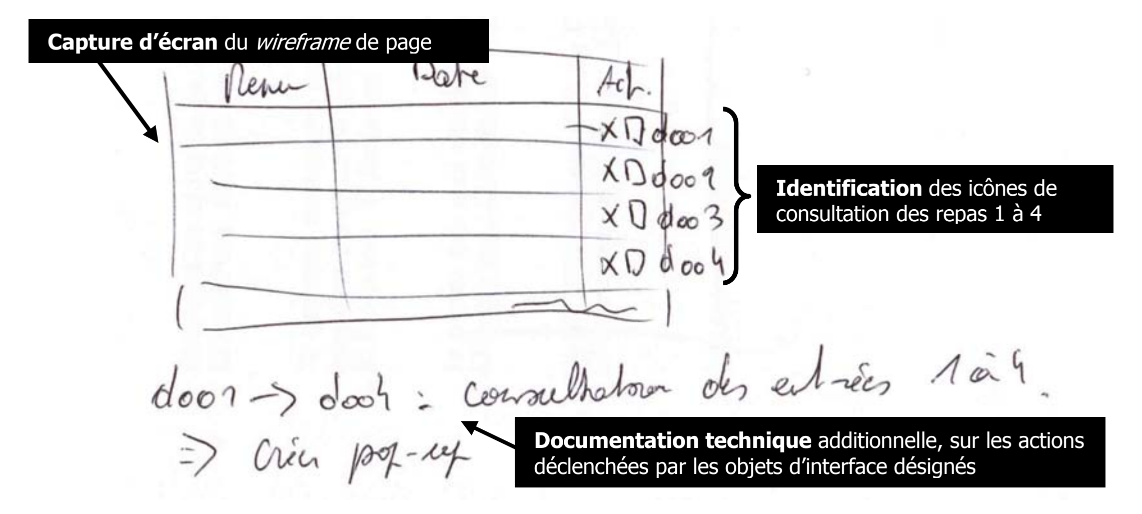


Figure 55. « *Requirements techniques* » tels que représentés par le sujet 8 dans son activité quotidienne

La Figure 55 montre un exemple de document technique produit par le sujet 8 lors de l'entretien consécutif à la réalisation du scénario de conception de modèle StateWebCharts. Elle illustre ses habitudes en matière de documentation de la spécification de l'interface. Chaque élément de l'interface est numéroté sur un *wireframe* de page ; dans une liste annexe, chaque élément est associé à l'information qu'il contient ou à l'action qu'il déclenche, si possible en termes d'accès à une base de données. Les sujets 4 et 7 font également allusion à des documents de ce type.

Lisibilité du formalisme StateWebCharts

Le formalisme StateWebCharts tel qu'il a été présenté aux sujets lors de l'évaluation était très simplifié par rapport à son pouvoir d'expression réel (cf. §5.4.4). Cette simplification avait pour but de recentrer l'analyste sur des problématiques purement d'interface, avec une spécification des enchaînements d'écrans tels que perçus par l'utilisateur de l'application Web. Cette simplification avait également pour but de fournir une notation lisible par tous les intervenants de la conception. Si les analystes ont tous déclaré que, avec leur connaissance de StateCharts, l'acquisition du formalisme StateWebCharts avait été immédiate, ils ont toutefois fait remarquer que ce modèle ne serait pas présentable au client sans explications de leur part.

« Peut être que les *states* devraient être plus représentatifs, que les noms définissent mieux les... Parce que, bon, au niveau des symboles, ce n'est pas très compliqué donc je crois que ça va, quoi… Mais au niveau des noms des *states*, il faudrait que ce soit un peu plus réfléchi. »
-- Sujet 2, le 16 avril 2008.

« Tous les intervenants n'ont certainement pas la connaissance, en tous cas, de StateWebCharts… Des diagrammes d'activité non plus… C'est surtout ça en fait… Parce que bon, une page, c'est très facile, beaucoup de gens ont Internet… Les *screenflows*, ça c'est plus compliqué (…) C'est compréhensible mais je pense que si la personne n'a jamais vu ça, elle va être un peu interloquée. [Il faut une phase d'explication], mais ça va très vite, c'est pour ça que je mets "plutôt oui" [à la question "Est-ce lisible et critiquable par chacun des intervenants de la conception ?"] »

-- Sujet 3, le 16 avril 2008.

« Le prototype [des pages] serait plus parlant pour le client que [le modèle StateWebCharts]. Il pourrait le lire, il pourrait le comprendre, mais je pense que c'est plus convivial pour lui de [discuter] le prototype, et que [le modèle StateWebCharts], c'est plus un support pour les programmeurs, ou en interne. »
-- Sujet 6, le 17 avril 2008.

« [Certains de nos clients] sont de la vieille école et… Ils ne comprennent pas, par exemple, que c'est une maquette, que ce n'est pas à ça que ça ressemblera… Ce n'est pas un problème en soi, mais c'est en cela que [le modèle StateWebCharts] n'est pas spécialement lisible. »
-- Sujet 7, le 15 avril 2008.

Les sujets 2, 3 et 6 pensent que, moyennant des intitulés clairs et un accompagnement dans l'appréhension du formalisme, les modèles StateWebCharts pourraient même être lus par les clients des projets. Les sujets 3 et 7 pensent que les modèles StateWebCharts sont difficiles à lire car ils ne reflètent pas la réalité d'une application Web, pour un client peu connaisseur des technologies Web.

5.6. Interprétation et perspectives

Les variables expérimentales énoncées dans la définition des hypothèses de la section 5.3 sont récapitulées dans le Tableau 8 ci-dessous, telles qu'elles ont été évaluées en moyenne au fil des expérimentations. Au regard de ces variables, nous pouvons tirer les conclusions suivantes quant aux hypothèses émises dans la section 5.3 :

- **Hyp.1 : le StyleGuide et un formalisme de description de la structure hypertexte (p.ex. SWC) sont utilisables et ont un pouvoir expressif satisfaisant pour la conception d'interface.** L'utilisation du StyleGuide est très marginale, au profit des formes Visio, mieux connues des analystes rencontrés. L'utilisabilité de StateWebCharts comme formalisme de modélisation de la navigation est indéniable pour ces sujets familiers du langage StateCharts. Quant au pouvoir d'expression fourni, les analystes l'estiment correct pour une spécification à destination des développeurs, moyennant des annotations sur le modèle de navigation, intégrant les éléments d'interface déclencheurs de transitions.
- **Hyp.2 : les wireframes de page et le modèle de structure hypertexte aident au consensus.** Ce point est validé par les suggestions pertinentes et précises sur les productions étalon présentées aux sujets. La culture en utilisabilité des sujets s'avère de bonne qualité, les conseils qui leur ont été délivrés par le centre de compétences en utilisabilité sur des projets précédents sont mémorisés et intégrés.
- **Hyp.3 : le StyleGuide favorise la réutilisation de la connaissance issue d'expériences de conception.** Les problèmes de conception couverts par le StyleGuide ne sont pas cernés par les sujets, ce qui constitue un premier frein à leur implication dans l'évolution du StyleGuide : ils ne sont pas sûrs de la présence ou de l'absence d'un problème tant qu'ils n'ont pas contacté le centre de compétences en utilisabilité. La participation à l'évolution du StyleGuide est une perspective en-

thousiasmante pour les sujets, mais irréaliste sur la plupart des projets où leurs activités sont minutées.

Tableau 8. Variables expérimentales évaluées

	VARIABLES EXPÉRIMENTALES					
SCÉNARIO 1 Conception de *wireframe*	Utilisation du StyleGuide	Appropriation des formes Visio	Satisfaction	Adéquation	Adhésion à la méthode	
	+	++	+++	+++	++++	
SCÉNARIO 1BIS Conception SWC	Connaissance StateCharts	Habitudes en modélisation hypertexte	Utilisation optimale de SWC	Satisfaction vis-à-vis de SWC		
	+++	+++	+	+++		
SCÉNARIO 2 Discussion	Lisibilité	Culture en utilisabilité	Recul et souplesse			
	+++	+++	+++			
SCÉNARIO 3 Suggestion	Indices d'absence	Ressources utilisées	Aisance	Sentiment gratifiant	Implication	Motivation
	+	-	++	++++	++	+++

LÉGENDE ++++ Très bonne +++ Bonne ++ Faible + Très faible - Nul

La section 5.1 présentait les similarités entre notre méthode et celle de SmalS incluant le StyleGuide. Ces similarités ont permis qu'au travers de l'évaluation du StyleGuide, nous dégagions des axes d'approfondissement de notre méthode, et confortions certains des choix que nous avons faits. En dépit du faible nombre de participants à l'évaluation et de leur manque de pratique du StyleGuide, des pistes d'amélioration nettes ont en effet pu être dégagées. Elles sont présentées ci-après, d'abord sous formes de recommandations pour une utilisation plus optimale du StyleGuide au sein de SmalS (cf. §5.6.1), puis sous forme de perspectives pour l'évolution de notre propre méthode (cf. §5.6.2).

5.6.1. Optimisation du StyleGuide

Au vu des résultats collectés, les utilisateurs du StyleGuide sont prêts à participer à l'évolution de son contenu, moyennant des outils qui les guident dans cette participation. Cet aspect est à encourager en priorité dans la mesure où, comme le fait remarquer l'un des sujets, *Si le StyleGuide n'évolue pas, il ne sera plus utilisable*. Vient ensuite le problème d'utilisabilité du StyleGuide, relevé par nombre de sujets, et observé notamment au travers du très faible niveau d'utilisation du StyleGuide lors de l'évaluation (cf. §5.5.2). Une autre observation que nous avons pu dégager est le besoin d'une communication plus claire au niveau du StyleGuide, de ses évolutions, de ses nouvelles versions. Une communication régulière sera d'autant plus primordiale que les utilisateurs se sentiront investis dans la participation à l'évolution du StyleGuide, et vice versa.

Encourager la participation des utilisateurs du StyleGuide

La mise à disposition d'un formulaire de suggestion d'ajout ou de modification de *guidelines* est le premier pas, peu coûteux, de l'encouragement à la participation des utilisateurs à l'évolution du StyleGuide. Ce formulaire doit être accessible depuis toute page du StyleGuide, par exemple par un lien *How can I contribute?*[1] ou *Suggest an update*[2] dans l'en-tête, aux côtés du lien permettant d'imprimer la page affichée. Un suivi de ces suggestions doit être assuré, ne serait-ce que par un bref retour informel, par courrier électronique, du devenir de la suggestion à chaque étape de son étude : En cours d'étude, Écartée, Acceptée. Lors de l'acceptation d'une suggestion, sa prise en compte dans le StyleGuide doit être mise en avant. Nous suggérons un formulaire à remplir, dont les informations sont enregistrées dans une base de données et diffusées via une page Web. Un outil d'aide à la conception d'enquêtes de type LimeSurvey[3] peut

[1] Comment puis-je contribuer ?
[2] Suggérer une mise à jour
[3] http://www.limesurvey.org/ (dernier accès 25 sept. 08)

être utilisé comme support, comme nous l'avons utilisé dans notre catalogue de patrons[1] (cf. §4.3). Ce dernier, formulaire de suggestion d'ajout à notre catalogue, peut servir de base pour la conception du formulaire de suggestion pour le StyleGuide, mais les intitulés doivent correspondre aux titres et sous-titres utilisés dans le StyleGuide pour conserver les repères de l'utilisateur.

Au vu des remarques faites par les sujets observés, le formulaire de suggestion doit être accessible et accommodant. Les intitulés des champs doivent être détaillés et complétés par un exemple illustratif. Il doit également bien être spécifié que le formulaire peut être rempli partiellement : la forme du formulaire doit le suggérer très fortement. Si l'utilisabilité du formulaire n'est pas suffisamment considérée, les utilisateurs, très accaparés par leurs activités quotidiennes de travail, ne feront pas l'effort de le remplir. Il serait également appréciable qu'un budget transversal dédié à la participation au StyleGuide soit envisagé, au moins pour les analystes. Il est estimé *a priori* à 0,75 jour par mois et par analyste, et couvre la consultation régulière des actualités du StyleGuide et les plus rares suggestions d'ajout ou de modification.

Améliorer l'utilisabilité du StyleGuide

Avant tout, pour assurer des repères au lecteur et utilisateur du StyleGuide, nous suggérons de systématiser la forme des *guidelines* : chacun doit disposer de titres et sous-titres similaires, quelque soit le *guideline* considéré. Ces titres et sous-titres seront repris pour la conception du formulaire de suggestion. Les titres généralement rencontrés dans la littérature sont, en Anglais : *Title, Description, Rational, Examples (good and bad), Use cases (must / should / shouldn't / mustn't), Related links.* Les illustrations utilisées comme exemples sont d'autant plus signifiantes pour l'utilisateur qu'elles appartiennent à son domaine de conception, et notamment ici, de façon optimale, qu'elles sont des réalisations de l'entreprise elle-même. Sur les *guidelines* établis sur une base empirique, des retours sur expérience informels aideraient encore à la compréhension de ces illustrations.

Un moteur de recherche est en cours de déploiement pour une publication sur une version prochaine du StyleGuide, ce qui permettra aux utilisateurs d'accéder directement au *guideline* traitant de l'élément recherché. Un thésaurus est également en cours de développement : la combinaison du thésaurus avec le moteur de recherche permettrait d'enrichir encore la recherche de *guidelines*, en suggérant des associations de patrons.

La fonctionnalité par vues est appréciée des analystes qui sont utilisateurs expérimentés du StyleGuide : ils y voient une classification des informations selon l'étape du processus de conception (cf. §5.5.2). Avant tout, il serait bon d'éviter que, lorsque l'utilisateur change de vue, l'ensemble des menus de la colonne de gauche ne se replie. Cette caractéristique pose un problème de repérage, et pousse les utilisateurs à conserver la vue générale. Corriger ce problème permettrait un passage plus souple d'une vue à l'autre. Nous suggérons, par la suite, d'autoriser l'utilisateur à paramétrer les rôles pour lesquels il veut voir les *guidelines*, par exemple « Analyste fonctionnel » et « Graphiste ».

Améliorer la communication autour du StyleGuide

Tous les sujets interrogés pensent que leur expérience est utile à leurs collègues, et trouvent gratifiant de partager cette expérience. Le StyleGuide peut être un support de communication pour une communauté de concepteurs aux intérêts convergents, chacun ayant son rôle à jouer dans le bon fonctionnement des activités des autres. Cependant, les sujets interrogés n'utilisaient pas tous la même version du StyleGuide, et ce manque de cohérence est un obstacle à leur participation active à ce projet. L'équipe responsable du StyleGuide doit ainsi fournir un effort sur la communication autour du StyleGuide. Une nouvelle version doit être valorisée, avec un courrier électronique l'annonçant officiellement à tous les utilisateurs, et indiquant

[1] http://fpontico.free.fr/limesurvey/index.php?sid=72227&lang=fr (dernier accès : 25 sept. 08)

quelles avancées elle représente. Pour des versions comportant des changements importants, des réunions de présentation peuvent être l'occasion de lister ces changements sans entrer dans le cadre d'une formation formelle coûteuse en temps.

Les ajouts et modifications significatives de *guidelines* doivent également être mis en valeur. D'une part, ils sont le miroir de la vitalité du StyleGuide et donnent l'image d'une communauté active, qui donne envie aux utilisateurs de s'y investir. D'autre part, elles facilitent l'utilisation du StyleGuide : l'état du StyleGuide est mieux connu de l'utilisateur, ce qui est l'une des règles de base de l'utilisabilité de tout système. Ces annonces peuvent être faites sur la page d'accueil du StyleGuide, à chaque modification, et classées par ordre chronologique descendant (la plus récente d'abord). Un code couleur peut distinguer les modifications mineures (p.ex. correction de coquille, correction d'un bug sur une forme), des modifications de fond (p.ex. ajout d'une catégorie) ou des ajouts de *guidelines*. Les modifications significatives peuvent être, en plus, envoyées par courrier électronique, une fois par mois, aux utilisateurs du StyleGuide.

5.6.2. Bilan et perspectives pour notre méthode

Les observations effectuées sur le StyleGuide valident certains partis pris de notre méthode et suggèrent des perspectives de travail. Le déploiement d'une méthode de conception d'interfaces basée sur des patrons peut être rendue plus optimal d'après les observations relevées chez SmalS, et nous les présentons ci-après. Le formalisme StateWebCharts doit de plus être utilisé de façon plus rigoureuse, au vu des observations des sujets.

Rendre le catalogue de patron utilisable

Les atouts et les problèmes d'utilisabilité relevés sur le StyleGuide lors de l'expérimentation avec ses utilisateurs semblent être couverts par notre catalogue. Cette couverture devrait cependant franchir l'épreuve de l'expérimentation avec des utilisateurs réels pour être établie fermement. Nous proposons, tout comme le StyleGuide, une classification des patrons par niveau de granularité pour que soit respecté l'ordre naturel de réflexion dans les activités de conception : cette classification emporte l'adhésion des sujets interrogés (cf. §5.5.2). Quant au moteur de recherche suggéré par quelques sujets, il est intégré à notre catalogue grâce à MediaWiki que nous utilisons comme support. L'intégration d'une ontologie, prévue dans les perspectives de notre méthode (cf. §4.2.2) permettrait de couvrir les problèmes de terminologie rencontrés par les sujets. Ces derniers ne formulent en effet pas toujours les problèmes de conception de la manière dont ils sont formulés dans le catalogue de patrons (cf. §5.5.2)

Déployer une telle méthode en entreprise

Pour encourager la participation active des utilisateurs du catalogue, un budget transversal dédié à la participation à l'évolution du catalogue est à prévoir, à hauteur de 0,75 jour par mois pour tout utilisateur du catalogue. Le catalogue n'est utile que s'il est mis à jour, et l'équipe de gestion du catalogue a besoin de retours d'expérience venant des utilisateurs du catalogue eux-mêmes. Ce budget transversal comprend également un coup d'œil régulier aux actualités du catalogue, mises en avant sur la page d'accueil de notre catalogue (cf. Figure 38). De plus, tous les trois mois, l'envoi d'un courrier électronique est préconisé, courrier adressé à tous les utilisateurs du catalogue, pour les informer des ajouts et des modifications majeures effectuées. Sans une officialisation de ces activités de participation, le catalogue est voué à l'abandon de ses utilisateurs, faute d'informations à jour qu'ils puissent utiliser.

Intégrer des éléments de l'interface aux modèles StateWebCharts

Les éléments définis par les sujets comme manquants au formalisme StateWebCharts peuvent être représentés si le formalisme est considéré de façon plus complète. Il s'agit principalement de pré conditions sur les transitions, et de signalement de la partie de l'interface qui

permet l'activation d'une transition. Formaliser les transitions en explicitant les éléments d'interfaces réactifs peut aider à retrouver les informations requises dans les habitudes de travail des sujets interrogés. L'utilisateur de StateWebCharts qui souhaite modéliser la structure hypertexte de son application doit ainsi être incité à utiliser au maximum les stéréotypes des transitions. Il peut ainsi indiquer, sans entrer dans les détails des éléments d'interface qui seront fixés plus tard, que telle transition est activée lorsqu'un élément du menu est cliqué.

Quant au signalement des parties de l'interface qui composent un écran, dont le niveau de granularité est à peu près celui de nos « Composants de base », cela peut être indiqué sur un *wireframe* complémentaire où les différentes composantes d'un écran de l'application sont identifiées. Ce *wireframe* particulier pourrait également faire figurer les objets manipulés, sur la base du modèle de données entité – relation réalisé. La Figure 56 ci-dessous représente une ébauche de ce type de *wireframe* tel qu'il pourrait être annoté, dans le cadre de l'étude de cas de la cantine de Goville, pour la gestion des repas (cf. §4.2). Les spécifications en matière de contenu y apparaissent clairement : le contenu généré dynamiquement est mis en avant, sa localisation, son apparence, mais aussi la partie de la base de données qui permet de le récupérer.

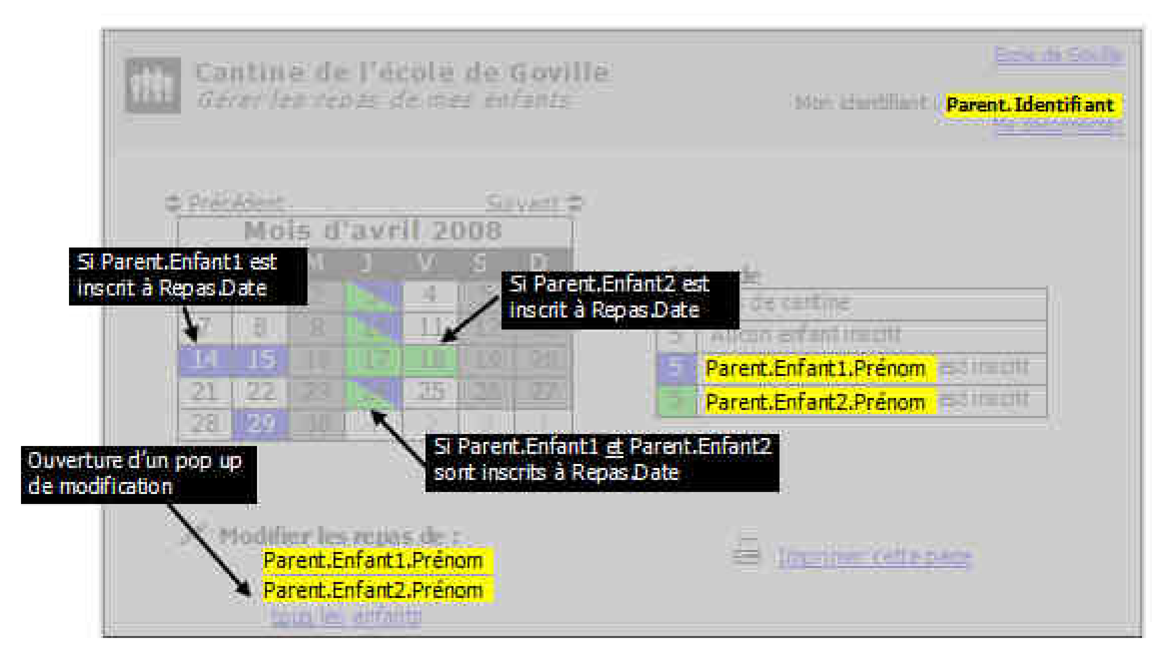


Figure 56. *Wireframe* d'identification des composants des écrans du site Web de la cantine de Goville

6. eGovIPM, un outil de support à la méthode

Résumé

Pour une intégration guidée et efficace de la méthode présentée dans le chapitre précédent, un outillage est fourni ici pour l'édition des prototypes d'enchaînements d'écrans. eGovIPM guide le concepteur dans l'instanciation de patrons d'enchaînements d'écrans, décrits avec le formalisme StateWebCharts. Une fois le modèle édité, un squelette de l'application est généré sur sa base, soit un ensemble de pages XHTML reliées entre elles conformément au modèle de navigation édité. Les *wireframes* personnalisés de page apparaissent sur les pages générées si le concepteur les spécifie ; sinon, ce sont les *wireframes* génériques issus du catalogue de patrons qui apparaissent. L'utilisation de l'outil eGovIPM est intégrée aux scénarios d'utilisation de la méthode et, quoiqu'il soit déjà opérationnel, plusieurs extensions sont envisageables à court terme pour que son utilisation couvre encore plus d'activités de conception.

6.1 Spécification des fonctionnalités

0 L'utilisateur type d'eGovIPM est un analyste en charge de la conception préliminaire de l'interface pour une application d'e-Procuration. Le modèle de navigation établi servira de support à la communication de l'analyste vers l'équipe de développement, en tant que définition de la navigation et des *wireframes* de pages. En effet, le client n'a généralement pas le recul nécessaire pour exprimer son avis et valider le modèle de navigation, qui est exprimé dans un jargon technique. Mais le client peut en connaissance de cause valider l'enchaînement d'écrans, et les *wireframes* qui y apparaissent, car le squelette d'application produit par eGovIPM offre un niveau d'interaction non réaliste mais suggestif. C'est en cela qu'eGovIPM est un premier pas dans la phase de développement : grâce à lui, l'analyste peut fournir une spécification de la navigation et de l'interface à l'équipe de développement.

Technologies utilisées

6.3 Scénario d'utilisation de l'outil eGovIPM

6.4 Perspectives

Le catalogue de patrons que nous avons mis à disposition comme support à l'analyse fonctionnelle d'une application d'e-Procuration contient notamment des patrons d'enchaînement d'écrans. Ces patrons contiennent des modèles de navigation génériques, formalisés grâce à StateWebCharts (Winckler et Palanque 2003). L'analyste est en charge de la personnalisation de ce modèle, de son adaptation au contexte particulier de conception. L'outil eGovIPM, acronyme d'*eGovernment Interface Patterns Manager*[1], est présenté dans ce chapitre. Il sert de support à la personnalisation des enchaînements d'écrans génériques présentés dans le catalogue et satisfait les besoins suivants :

- **Édition graphique** de modèles de navigation pour la personnalisation des modèles génériques fournis dans le catalogue de patrons, et la spécification de l'application. Chaque page du modèle de navigation en cours d'édition est typé : chacun correspond à un patron de page du catalogue. D'autres informations peuvent figurer sur une page (p.ex. titre, description). Pour une transition, il peut être précisé la condition de son activation ;

- **Analyse syntaxique** des modèles de navigation : vérification du respect de règles concernant l'enchaînement des écrans. Par exemple, pour un assistant à la réalisation de procédure, il peut être vérifié qu'une page de récapitulatif existe, et qu'elle vient à l'issue de la réalisation des étapes de la procédure ;

- **Génération d'un squelette de l'application** à partir du modèle édité, pour le support à la communication au cours de la conception. Il s'agit d'un ensemble de pages XHTML, liées entre elles conformément à la définition de la navigation donnée par le modèle de navigation édité. Chaque page fait figurer les informations qui ont été spécifiées à son sujet, ainsi que celles de ses transitions sortantes. Un exemple est montré dans la Figure 57.

Ce chapitre présente tout d'abord en détails les fonctionnalités de eGovIPM pour l'édition guidée de modèles de navigation et la génération d'un squelette de l'application (cf. §6.1). Puis

[1] Gestionnaire de patrons d'interfaces pour l'e-Gouvernement

les technologies utilisées pour l'implémentation de l'outil sont explicitées (cf. §6.2). Enfin, un scénario d'utilisation de l'outil illustre son mode d'utilisation (cf. §6.3) et des perspectives de développement de l'outil sont présentées (cf. §6.4).



Figure 57. Exemple de page XHTML générée par l'outil eGovIPM

6.1. Spécification des fonctionnalités

Notre catalogue de patrons met à disposition des enchaînements d'écrans génériques pour l'analyse fonctionnelle en e-Procuration. L'outil eGovIPM offre un support à la personnalisation de ces enchaînements d'écrans. La couverture du catalogue par l'outil eGovIPM est consignée dans l'Annexe F : il y figure l'ensemble des patrons qui sont supportés par l'outil, et les raisons techniques pour lesquelles certains patrons d'enchaînements d'écrans n'ont pas été retenus dans cette première version de l'outil.

eGovIPM est un plugin de l'environnement Eclipse (Eclipse 2008a) qui guide le concepteur dans l'instanciation de patrons, dans le respect de contraintes d'édition définies (cf. §6.1.1). Le concepteur peut ensuite, grâce à eGovIPM, générer le squelette de l'application dont il a spécifié le modèle de navigation (cf. §6.1.2).

6.1.1. Instanciation de patrons

L'instanciation de patrons d'enchaînements d'écrans consiste en la modification d'un patron d'enchaînement d'écrans générique tel qu'il est fourni dans le catalogue de patrons. Les enchaînements d'écrans sont formalisés grâce à StateWebCharts (Winckler et Palanque 2003) à partir des enchaînements d'écrans génériques suivants (pour la version de eGovIPM en date de juin 2008) :

- Tableau de bord ;
- Assistant avec guidage fort ;
- Assistant avec guidage souple ;
- Assistant avec récapitulatif éditable ;

- Libre : aucune contrainte n'est associée à l'édition de ce modèle.

Les modifications consistent en l'ajout, l'édition ou la suppression d'une page ou d'une transition. Les éléments sont ajoutés depuis la palette qui est reproduite sur la Figure 58 : neuf pages et un enchaînement d'écrans peuvent ainsi être ajoutés à un modèle en cours d'édition.

Il est à noter qu'eGovIPM est une extension de SWCEditor (Winckler et coll. 2004), éditeur et simulateur de modèles StateWebCharts. Par conséquent, il est également possible de créer et d'éditer des diagrammes SWC via la palette d'édition d'eGovIPM. Naturellement, la génération du squelette de l'application n'est possible que si les objets eGovIPM ont été utilisés.

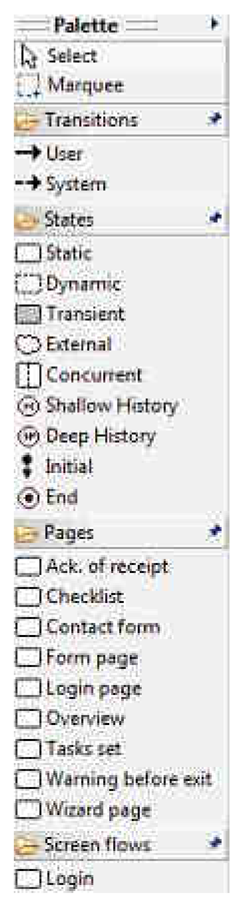


Figure 58. Palette de l'outil eGovIPM

L'utilisateur d'eGovIPM peut aussi éditer les informations du modèle de navigation, par interaction directe avec un objet (p.ex. double clic sur un état pour modifier son label) ou par le biais de la vue Propriétés de l'environnement Eclipse qui est en partie reproduite sur la Figure 59 : la page « Gestion de repas » est sélectionnée sur cet exemple et la vue Propriétés permet de voir, entre autres informations, son titre, son type (correspondant à une entrée dans le catalogue de patrons, ici « Liste de tâches »), une description succincte de son contenu, ou encore le *wireframe* qui lui est associé, sous l'attribut image.

Pour ce qui concerne le *wireframe* d'une page, dans le cas de la Figure 59, l'utilisateur a spécifié, en éditant directement le champ image dans la vue Propriétés, que le *wireframe* de cette page figurait dans le fichier cantine-calendrier.jpg. L'utilisateur aura pris soin d'enregistrer son *wireframe* à un format image de type JPG, dans le dossier de *wireframes* personnalisé spécifié dans les préférences d'eGovIPM. Si le champ est laissé vide pour une page *P*, le *wireframe* associé est le *wireframe* générique du type de page de *P*, issu du catalogue de patrons (p.ex. pour la page « Gestion de repas », le *wireframe* du patron de page « Liste de tâches » apparaît si le champ image est laissé vide).

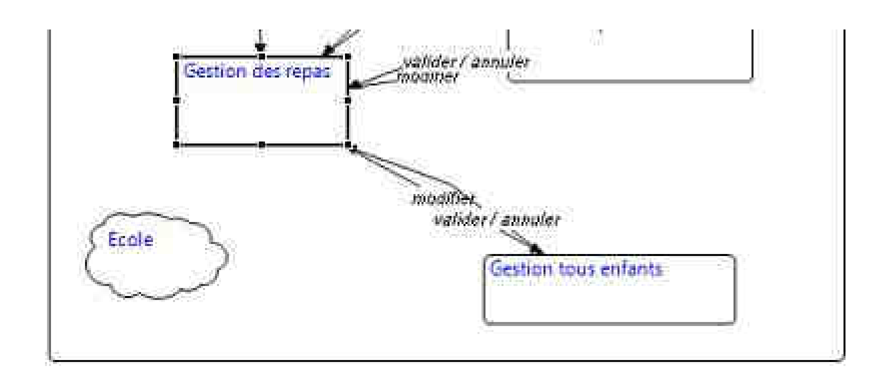


Figure 59. Vue Propriétés de eGovIPM

Notre contribution ne se situe pas au niveau du contenu des recommandations qui sont dispensées. Nous avions déjà évoqué cette limite de notre travail à propos du contenu du catalogue (cf. §4.5), elle est aussi valable pour la définition des contraintes qui régissent l'édition de modèles de navigation dans eGovIPM. Les contraintes que nous avons établies n'ont en effet pas fait l'objet de validation par un expert. L'objectif est ici de montrer la faisabilité de l'édition d'un modèle de navigation qui soit contrainte par un ensemble de règles pour lesquelles nous proposons ce mode de description simple.

Une page ou un enchaînement d'écrans peut être ajouté au modèle de navigation en cours d'édition si son type l'y autorise. Le Tableau 9 montre le nombre d'apparitions autorisées dans un enchaînement d'écrans pour une page ou un enchaînement d'écrans. Le tableau se lit comme suit : l'enchaînement d'écrans qui figure sur une ligne peut contenir *x* fois l'élément qui figure sur une colonne. Par exemple, un tableau de bord peut contenir un formulaire de contact mais pas d'étape d'assistant.

Tableau 9. Nombre d'apparitions autorisées dans un enchaînement d'écrans

	Accusé de réception	Liste d'informations à rassembler	Formulaire de contact	Page de formulaire	Page de login	Récapitulatif	Liste de tâches	Avertissement avant sortie	Étape d'assistant	Séquence de login
Tableau de bord	0	0	1	2..*	0	0	1	1	0	1
Assistant avec guidage fort	1	1	1	0	0	1	0	1	2..*	1
Assistant avec guidage souple	1	1	1	0	0	1	0	1	2..*	1
Assistant avec récapitulatif éditable	1	1	1	0	0	1	0	1	2..*	1
Libre	*	*	*	*	*	*	*	*	*	*

L'édition d'une transition consiste en la redirection d'une extrémité de la transition d'un état vers un autre. Les contraintes sur ce type d'édition sont présentées dans le Tableau 10.

Tableau 10. Transitions autorisées d'une page à une autre page

Sens de la transition	Accusé de réception	Liste d'informations à rassembler	Formulaire de contact	Page de formulaire	Page de login	Récapitulatif	Liste de tâches	Avertissement avant sortie	Étape d'assistant	Séquence de login
Accusé de réception	x	v	v	x	x	x	x	v	x	v
Liste d'informations à rassembler	x	x	v	x	x	X	x	v	v	v
Formulaire de contact	v	v	x	x	x	X	v	v	x	v
Page de formulaire	x	x	v	x	x	X	v	x	x	x
Page de login	x	x	x	v	x	X	x	x	x	x
Récapitulatif	v	x	v	x	x	v Si assistant avec recapitulatif éditable x Sinon	x	v	v	v
Liste de tâches	x	x	v	v	x	X	v	v	x	x
Avertissement avant sortie	x	v	x	x	x	X	v	x	x	x
Étape d'assistant	x	x	v	x	x	v Si dernière étape x Sinon	x	v	v	x
Séquence de login	x	v	v	v	x	X	v	x	x	x

6.1.2. Génération d'un squelette de l'application

À partir du modèle StateWebCharts représentant l'enchaînement d'écrans, édité avec l'outil eGovIPM comme spécifié dans la section 6.1.1, un squelette de l'application à concevoir peut être généré automatiquement. L'Annexe G montre la manière dont le modèle StateWebCharts est exploité pour créer un ensemble de pages XHTML qui soient reliées entre elles conformément au modèle StateWebCharts. Sur la base d'un modèle de navigation StateWebCharts, il est ainsi possible de constituer un ensemble de pages XHTML, liées entre elles conformément au modèle de navigation, et comportant un ensemble d'informations. La Figure 60 montre un exemple de page générée, mettant en avant les informations qu'elle contient, extraites du modèle StateWebCharts édité dans eGovIPM.

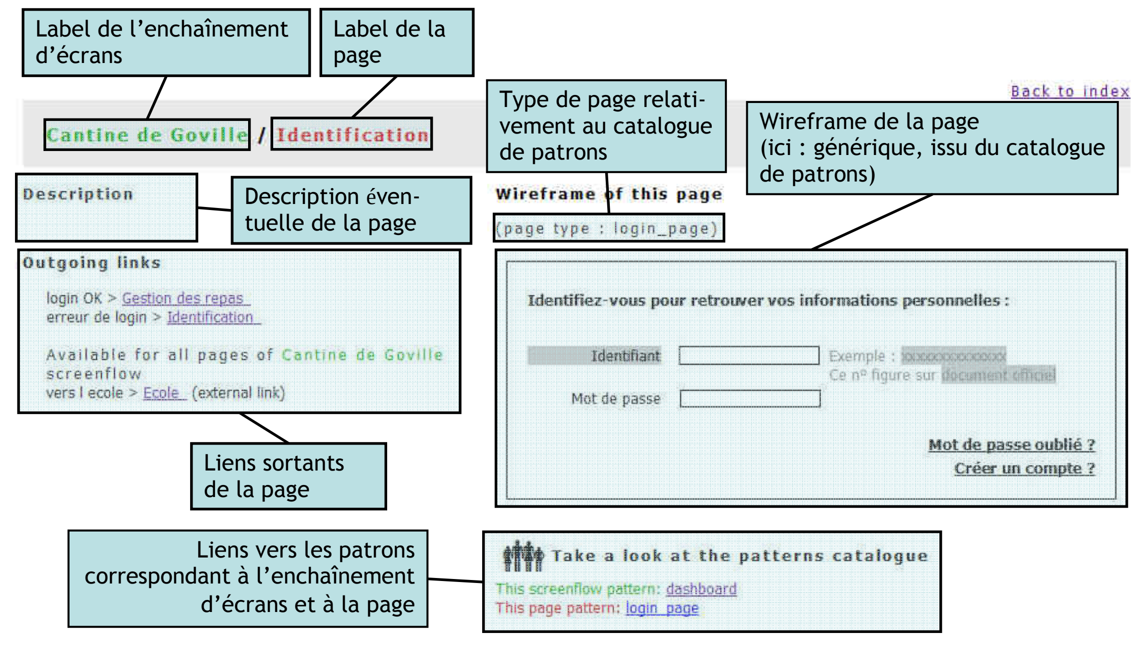


Figure 60. Exemple de page XHTML générée automatiquement

L'utilisateur type d'eGovIPM est un analyste en charge de la conception préliminaire de l'interface pour une application d'e-Procuration. Le modèle de navigation établi servira de support à la communication de l'analyste vers l'équipe de développement, en tant que définition de la navigation et des *wireframes* de pages. En effet, le client n'a généralement pas le recul nécessaire pour exprimer son avis et valider le modèle de navigation, qui est exprimé dans un jargon technique. Mais le client peut en connaissance de cause valider l'enchaînement d'écrans, et les *wireframes* qui y apparaissent, car le squelette d'application produit par eGovIPM offre un niveau d'interaction non réaliste mais suggestif. C'est en cela qu'eGovIPM est un premier pas dans la phase de développement : grâce à lui, l'analyste peut fournir une spécification de la navigation et de l'interface à l'équipe de développement.

6.2. TECHNOLOGIES UTILISÉES

eGovIPM est un plugin Java de l'environnement de développement Eclipse, ce qui assure une portabilité et une facilité d'installation de l'outil. Les modèles StateWebCharts manipulés sont stockés au format XML et édités de façon interactive avec une palette d'objets (pages et enchaînements d'écrans) mise à disposition de l'utilisateur, comme il apparaît à droite de la Figure 61. Cette section présente l'architecture logicielle de l'outil eGovIPM (cf. §6.2.1), puis expose les aspects techniques de ses fonctionnalités d'édition (cf. §6.2.2) et de génération automatique (cf. §6.2.3).

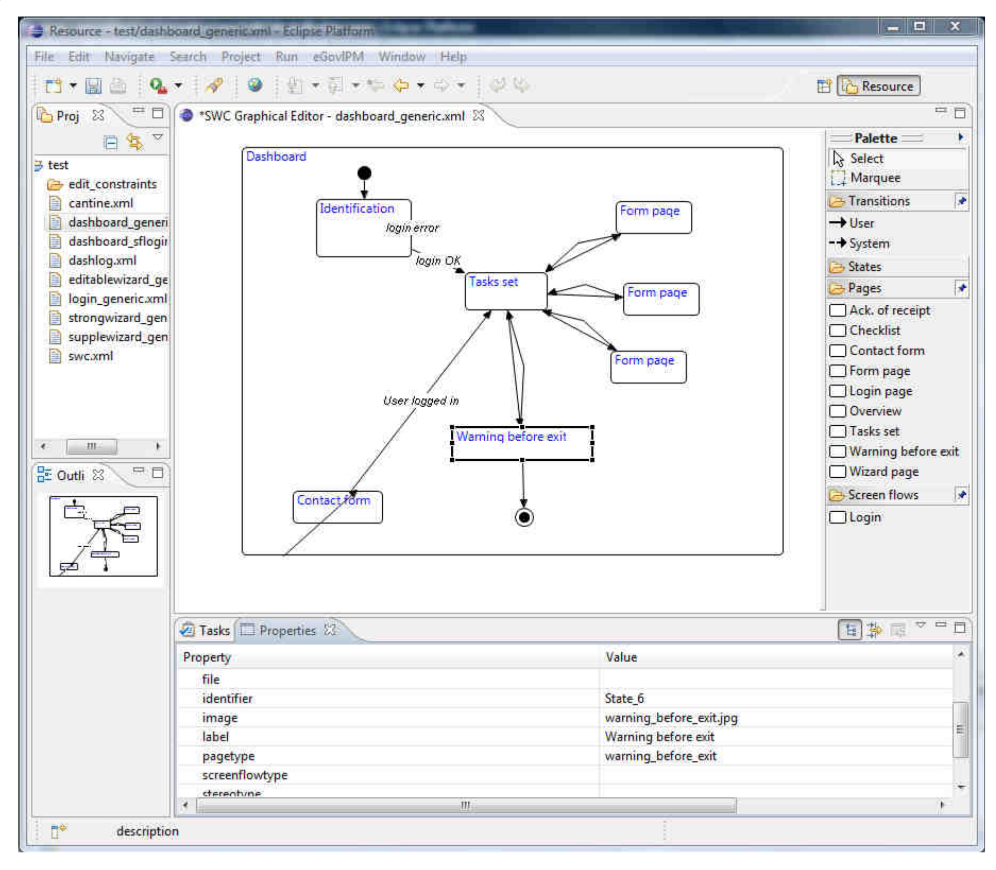


Figure 61. eGovIPM, un plugin d'Eclipse

6.2.1. Architecture logicielle

Aujourd'hui, Eclipse (Eclipse 2008a) est une plateforme de développement robuste et incontournable qui offre des fonctionnalités avancées d'édition graphique, de gestion de l'espace de travail, d'analyse automatique, etc. En outre, cette plateforme peut être étendue selon ses besoins grâce à l'ajout de plugins (p.ex. environnement de développement C++, éditeur graphique UML, etc.). Une nouvelle version de SWCEditor (Winckler et coll. 2004) a été développée

sous forme de plugin Eclipse pour s'intégrer à cette plateforme en conservant les fonctionnalités d'édition et de simulation de modèles de navigation StateWebCharts fournies par SWCEditor.

Notre outil, eGovIPM, est une extension de cette nouvelle version de SWCEditor, dont l'objectif est de satisfaire les besoins d'opérationnalisation de notre méthodologie. Pour cela, nous avons implémenté deux modules principaux (cf. Figure 62) :

- Éditeur de modèles StateWebCharts : il permet l'édition graphique de modèles StateWebCharts, à partir de modèles génériques, et sous les contraintes d'édition que nous avons établies plus haut (cf. §6.1.1) ;
- Générateur du squelette de l'application : à partir d'un modèle StateWebCharts personnalisé, il permet de générer un ensemble de pages XHTML lui correspondant (cf. §6.1.2).

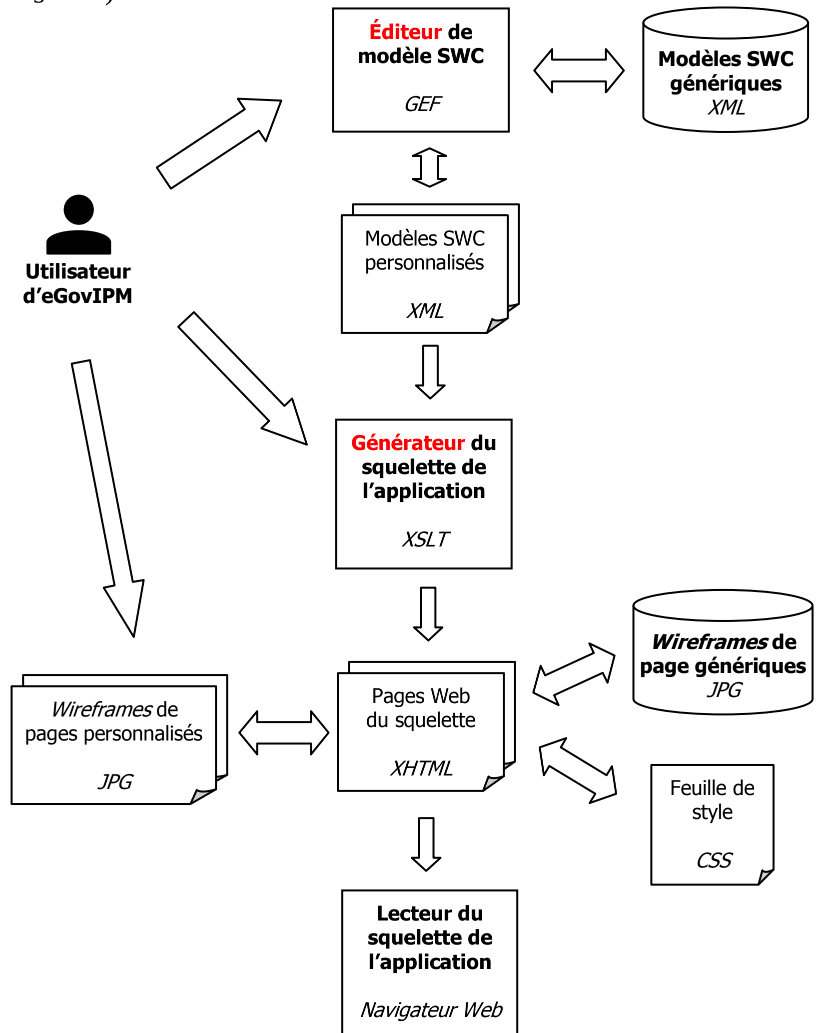


Figure 62. Architecture logicielle de eGovIPM

6.2.2. Édition de modèles StateWebCharts

L'interface de l'éditeur est montrée dans la Figure 61. La création des différents modèles se fait en ajoutant (par *drag & drop*) des éléments (pages ou enchaînements d'écran) depuis la palette (cf. Figure 58). GEF *Graphical Editing Framework*[1] (Eclipse 2008b) est la librairie Java qui nous a permis de gérer l'affichage des objets dans l'éditeur eGovIPM et l'offre d'interactions avec ceux-ci (sélection, création, connexion).

Une fois le modèle édité, l'utilisateur peut le sauvegarder. Les modèles StateWebCharts édités sous eGovIPM sont enregistrés au format XML *eXtensible Markup Language*[2] (W3C 2008a) pour formaliser leur structure et faciliter leur traitement. La persistance des modèles est gérée par la technologie JAXB *Java Architecture for XML Binding*[3] (Glassfish 2008). Un fichier XSD *XML Schema Description* (W3C 2008b) permet de définir la structure et la syntaxe des fichiers XML décrivant les modèles StateWebCharts : il est disponible dans l'Annexe H. La Figure 63 montre l'extrait d'un fichier XML : il représente un enchaînement d'écrans par le biais d'un modèle StateWebCharts pour l'exploitation dans eGovIPM. Les informations sémantiques sur le modèle (p.ex. identifiants des états et transitions, labels, types de pages et d'enchaînements d'écrans) sont mêlés à des informations de présentation (taille et positionnement des objets dans l'espace d'édition) sur l'élément GraphicBounds.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Swc xmlns= "http://liihs.irit.fr/swceditor">
   <CompositeState concurrent="false" file="" id="State_0" initial="false" label="Dashboard"
 screenflowtype="dashboard" stereotype="">
      <GraphicBounds height="385" width="510" x="183" y="97"/>
      <CompositeState concurrent="false" file="" id="State_l0" initial="true" label="Login"
stereotype="" screenflowtype="login">
         <GraphicBounds height="240" width="313" x="115" y="84"/>
         <StaticState file="" id="State_l1" initial="true" label="Login page" stereotype=""
type="static" pagetype="login_page" image="login_page.jpg">
            <GraphicBounds height="40" width="81" x="146" y="181"/>
         </StaticState>
         <StaticState file="" id="State_l2" initial="false" label="Form page" stereotype=""
type="static" pagetype="form_page" image="form_page.jpg" description="Request for
receiving password again">
            <GraphicBounds height="36" width="78" x="310" y="96"/>
         </StaticState>
         <StaticState file="" id="State_l3" initial="false" label="Form page" stereotype=""
type="static" pagetype="form_page" image="form_page.jpg" description="Fill in personal
data">
            <GraphicBounds height="36" width="78" x="283" y="266"/>
         </StaticState>
         <EndState id="State_lf">
            <GraphicBounds height="18" width="18" x="317" y="193"/>
         </EndState>
      </CompositeState>
      <StaticState file="" id="State_2" image="tasks_set.jpg" initial="false" label="Tasks
```

Figure 63. Exemple de fichier XML représentant un enchaînement d'écrans

6.2.3. Génération du squelette de l'application

Les pages XHTML composant le squelette de l'application sont générées automatiquement depuis un modèle StateWebCharts édité par l'utilisateur d'eGovIPM. Le langage XSLT *Extensible Stylesheet Language Transformations*[7] (W3C 2008c) permet de parcourir de façon

[1] Cadre d'édition graphique
[2] Langage de balisage extensible
[3] Architecture Java pour la liaison avec XML

[7] Transformations à partir d'un langage extensible de feuille de style

systématique les éléments d'un fichier XML et d'effectuer des opérations de transformation sur ces éléments.

Le modèle XSLT de transformation est reproduit partiellement sur la Figure 64. Ici, nous cherchons à transformer un fichier XML (celui qui représente le modèle de navigation State-WebCharts) en un ensemble de pages XHTML (une par état du modèle StateWebCharts), liées entre elles conformément aux informations contenues dans le fichier XML lui-même, et dont le contenu provient de ce même fichier XML.

La transformation s'opère par traitement de nœuds identifiés sur le schéma XML. La règle 1 (cf. Figure 64) concerne par exemple l'élément Swc qui est le nœud racine d'un fichier XML décrivant un enchaînement d'écrans sous eGovIPM (cf. exemple de fichier XML dans la Figure 63 ; schéma XSD grammaire des fichiers XML manipulés sous eGovIPM dans l'Annexe H). C'est la première règle qui sera exécutée dans le modèle XSLT implémenté pour eGovIPM. Il est décrit dans cette règle que, pour tout nœud Swc, la règle CompositeState doit être appliquée pour chacun des fils de type CompositeState (c'est-à-dire pour tout état composé, page ou enchaînement d'écrans). La règle 2 concerne justement les nœuds de type CompositeState (cf. Figure 64) : elle commande un traitement systématique pour les fils de ces nœuds, différenciant les fils :

- Dont l'attribut initial est vrai : la règle correspondant à leur nature est exécutée, avec le paramètre init ;
- De type StaticState : il s'agit des nœuds représentant une page de l'enchaînement d'écrans. La règle concernant les nœuds de type StaticState est exécutée avec le paramètre page ;
- De type EndState : il s'agit de l'état final du modèle de navigation. La règle concernant les nœuds de type EndState est exécutée avec le paramètre page.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:egovipm="http://liihs.irit.fr/swceditor"
  version="2.0">

<xsl:output name="html"
    method="html"
    indent="yes"
    doctype-public="-//W3C//DTD HTML 3.2 Final//EN"/>

Règle 1
<xsl:template match="egovipm:Swc">
    <xsl:apply-templates select="egovipm:CompositeState"/>
</xsl:template>

Règle 2
<xsl:template match="//egovipm:CompositeState">
        <xsl:apply-templates select="egovipm:*[@initial='true']" mode="init"/>
        <xsl:apply-templates select="//egovipm:StaticState" mode="page"/>
        <xsl:apply-templates select="//egovipm:EndState" mode="page"/>
</xsl:template>
```

Figure 64. Extrait du fichier de transformation XSLT

[2] Feuille de style en cascade

Ainsi, l'ensemble des noeuds fils du noeud racine Swc seront traités en cascade et de façon systématique. Des règles commanderont ainsi la création de fichiers XHTML, d'autres l'affichage d'un texte réactif correspondant à un lien sortant d'une page, etc, jusqu'à composer l'ensemble des pages XHTML du squelette de l'application.

Saxon (Kay 2008) est le moteur qui, déclenché depuis Eclipse, exécute effectivement la transformation XSLT. Un fichier CSS *Cascading StyleSheet*[2] (W3C 2008d) est lié à chacune des pages XHTML générées, pour composer leur apparence graphique.

6.3. Scénario d'utilisation de l'outil eGovIPM

Nous illustrons ici l'utilisation d'eGovIPM en reprenant l'étude de cas fictive utilisée comme support au scénario d'utilisation de la méthode (cf. §4.4) : la conception du site Web de la cantine de l'école de Goville.

6.3.1. Édition du modèle de navigation

Le modèle de navigation avait déjà été présenté lors de l'illustration de la méthode que nous proposons (cf. §4.4.1). Il est reproduit sur la Figure 66. Pour l'obtenir, voici la séquence d'actions qu'a réalisée l'utilisateur d'eGovIPM, en charge de la conception de la navigation du site Web de la cantine de Goville.

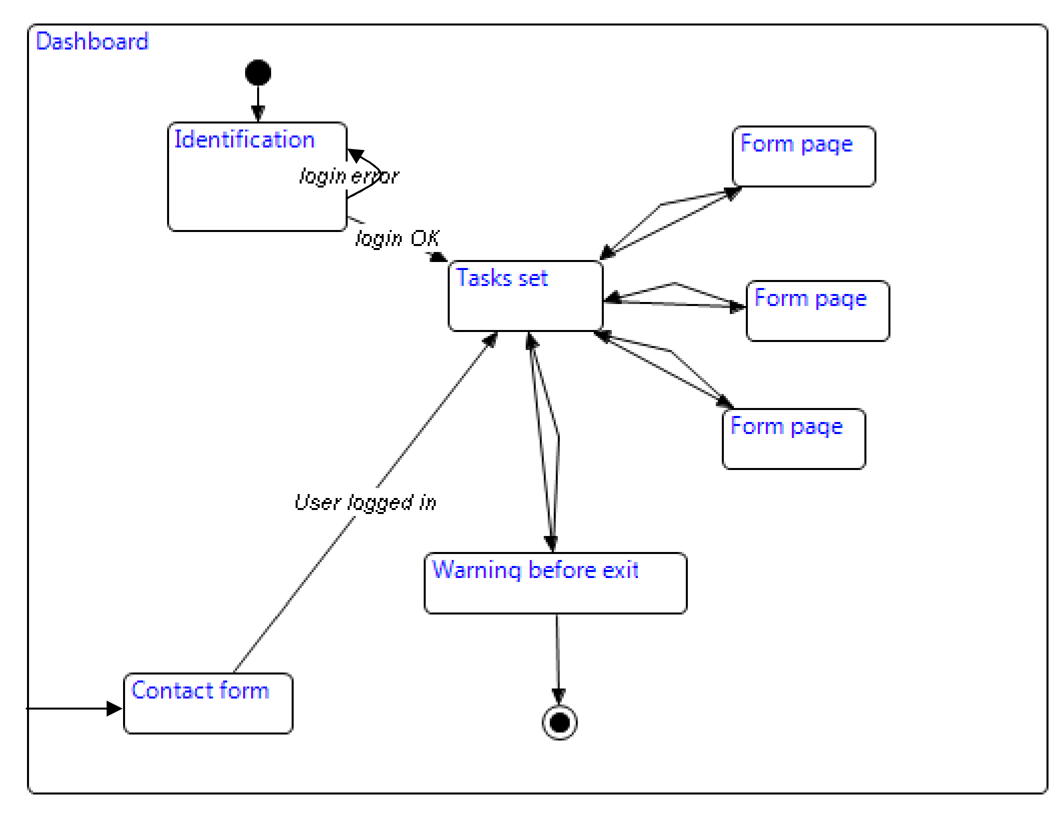


Figure 65. Modèle StateWebCharts de l'enchaînement d'écrans générique « Tableau de bord »

- Ouverture d'un nouveau modèle StateWebCharts dans eGovIPM, sur la base du modèle générique « Tableau de bord », modèle reproduit sur la Figure 65 ;
- Suppression des pages « *Contact form*[1] » et « *Warning before exit*[2] » que l'utilisateur ne trouve pas pertinentes dans son contexte : ces opérations sont autorisées ;

[1] Formulaire de contact
[2] Avertissement avant sortie

- Ajout d'un état externe StateWebCharts représentant le site Web de l'école : il spécifie l'adresse URL de l'école dans l'attribut file du volet Propriétés d'eGovIPM ;
- Ajout d'une transition vers la page d'identification, et d'une autre vers l'état externe représentant le site Web de l'école ;
- Édition des noms des états et des transitions ;
- Enregistrement au format JPG du *wireframe* personnalisé pour la page « Gestion des repas » dans le dossier spécifié par les préférences d'eGovIPM ; puis, édition de l'attribut image de la page « Gestion des repas » dans le volet Propriétés ;
- Repositionnement des états et des transitions pour optimiser la présentation du modèle.

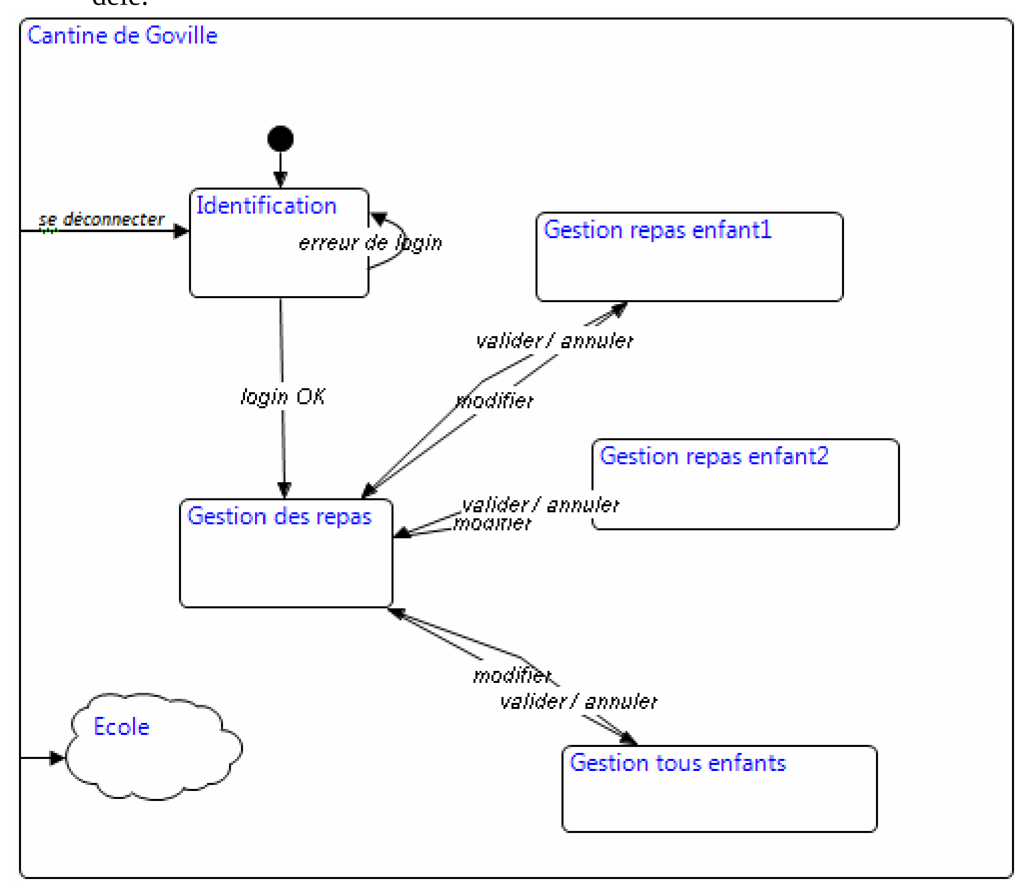


Figure 66. Modèle de navigation SWC du site Web de la cantine de Goville

6.3.2. Génération du squelette de l'application

À partir du modèle de navigation édité dans la section précédente (cf. Figure 66), l'utilisateur lance, depuis le menu d'eGovIPM, la génération du squelette de l'application dont il vient de spécifier l'enchaînement d'écrans. La Figure 67 montre les fichiers qui ont été créés pour la génération du squelette de l'application de la cantine de Goville. Ils sont créés dans un dossier dont l'emplacement est spécifié dans les préférences utilisateur d'eGovIPM. Chaque fichier porte comme nom l'identifiant de l'état StateWebCharts auquel il correspond (attribut id), par exemple State_20.html. Le fichier index.html contient une simple redirection vers celui de ces états qui est défini comme initial.

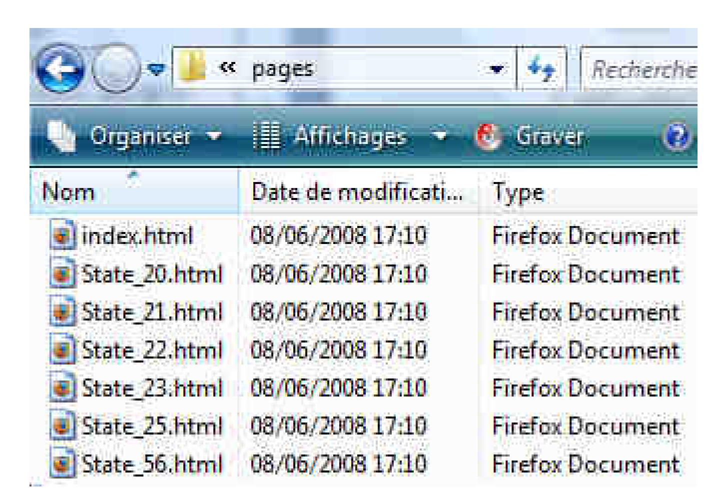


Figure 67. Fichiers créés par la génération du squelette de l'application

Les pages générées ne sont pas toutes reproduites ici. Seules apparaissent ci-après la page « Identification » (cf. Figure 68) et la page « Gestion des repas » (cf. Figure 69). Le *wireframe* de la page « Identification » n'a pas été personnalisé par l'utilisateur : le *wireframe* générique apparaît donc, tel qu'il figure dans le catalogue de patrons pour le patron « Page de login[1] ». Au contraire, le *wireframe* qui apparaît sur la page « Gestion des repas » est celui qui a été enregistré au format JPG par l'utilisateur et dont le nom a été indiqué sur le modèle StateWebCharts lors de son édition dans eGovIPM. Les liens hypertextes figurant dans ces pages sont actifs :

- Le lien « *Back to index*[2] » (dans le bandeau d'en-tête, à droite) apparaît sur toutes les pages du squelette ; il mène à la page dont l'état StateWebCharts correspondant est défini comme initial ;
- Les liens sortant de la page courante mènent vers d'autres pages. Les liens spécifiques à une page (c'est-à-dire les liens dont la source est l'état StateWebCharts correspondant à la page) sont distingués des liens accessibles depuis tout l'enchaînement d'écrans courant (c'est-à-dire les liens dont la source est le père de l'état StateWebCharts correspondant à la page) ;
- Parmi les liens sortants de la page, les liens externes sont étiquetés comme tels (cf. lien vers l'école de Goville marqué *external link*[3]). La cible de ces liens est l'URL indiquée dans l'attribut file de ces états externes. Pour conserver la main de la navigation, ces liens sont ouverts dans une nouvelle fenêtre.

[1] http://fpontico.free.fr/index.php?title=Login (dernier accès : 25 sept. 08)
[2] Retour à l'index
[3] Lien externe

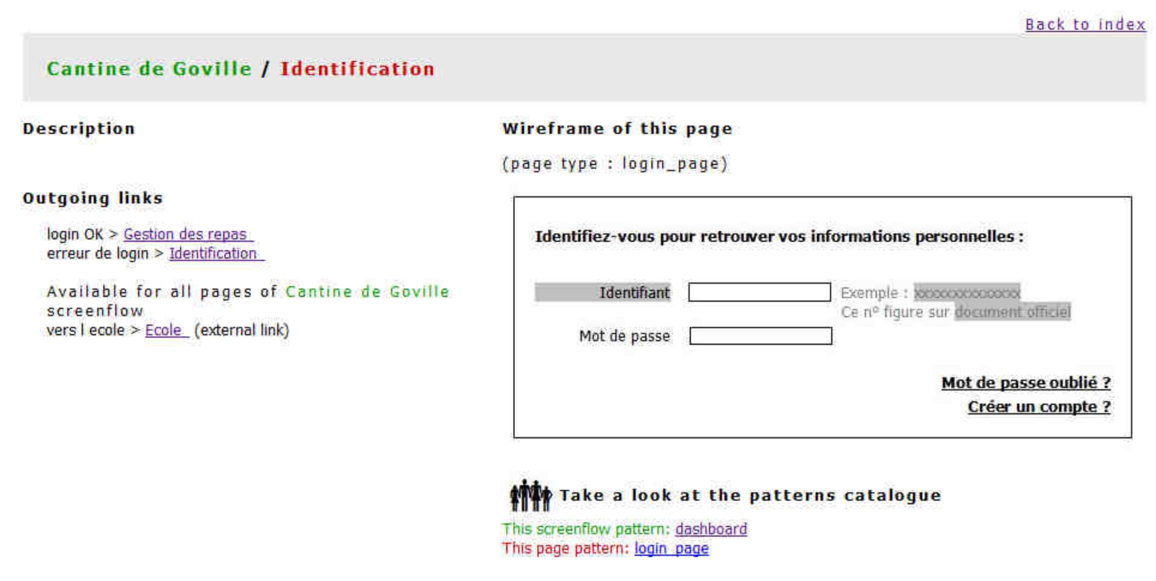


Figure 68. Page XHTML générée pour l' « Identification »

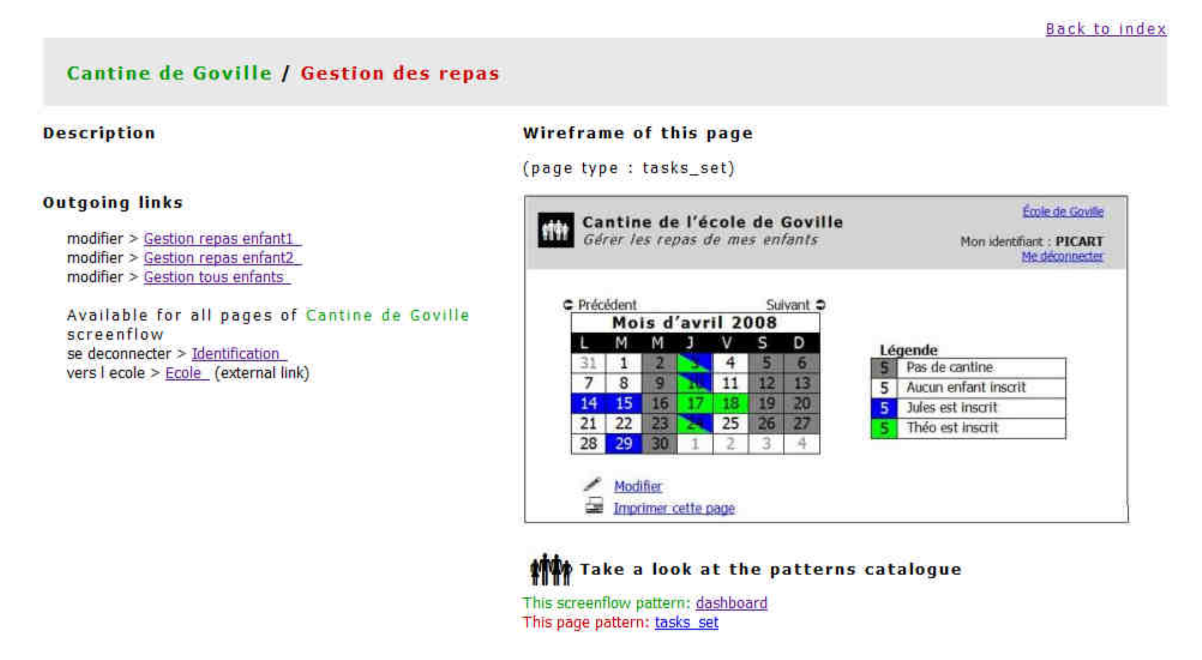


Figure 69. Page XHTML générée pour la « Gestion des repas »

6.4. Perspectives

L'outil eGovIPM offre un support à la spécification de la navigation d'une application d'e-Procuration, sur la base d'enchaînements d'écrans reconnus comme étant de bonnes pratiques de conception dans le domaine. Il offre aussi un moyen rapide de produire un squelette de l'application pour favoriser la communication de l'analyste avec le client mais aussi avec l'équipe de développement. L'utilité et l'utilisabilité de l'outil eGovIPM pourraient être considérablement améliorées par les extensions suivantes :

- **Ajout des enchaînements d'écrans à la palette** : les enchaînements d'écrans spécifiés comme génériques dans notre catalogue de patrons peuvent être combinés dans les applications complexes de type portail (cf. §4.2.2). Aujourd'hui, seul l'enchaînement d'écrans « Login » apparaît dans la palette d'objets pouvant être

ajoutés à un modèle de navigation ; les autres enchaînements d'écrans doivent pouvoir être ajoutés de même ;

- **Intégration de tous les enchaînements d'écrans du catalogue** : comme spécifié dans l'Annexe F, le patron « Gestion de dossiers » n'est pas couvert par l'outil eGovIPM ; en effet deux états concurrents peuvent être spécifiés au maximum dans SWCEditor ;
- **Formalisation et affinement des contraintes d'édition** : les contraintes d'édition du modèle figurent à l'heure actuelle dans le corps du code de l'application. Les extraire de ce code et les formaliser selon un langage d'édition de contraintes de type OCL *Object Constraint Language*[1] (OMG 2007) permettrait une meilleure gestion de celles-ci. Il serait envisageable de proposer leur édition aux utilisateurs expérimentés ;
- **Pédagogie autour des corrections fournies** : une correction préventive des contraintes d'édition rendrait l'édition de modèles de navigation plus utilisable (p.ex. griser les objets de la palette ne pouvant figurer dans le modèle en cours d'édition ; faire apparaître en rouge les pages et transitions ne pouvant être modifiés ; afficher dans une couleur différente les pages et transitions ajoutées par l'utilisateur). Lier une contrainte à une recommandation ergonomique qui s'afficherait en même temps que la contrainte est violée permettrait de transmettre à l'utilisateur des bases de conception centrée utilisateur ;
- **Insertion de fragments de code** : au lieu d'insérer une image du *wireframe* de la page comme élément principal de la page générée, du code XHTML pourrait être directement intégré. Ce code serait stocké dans le catalogue de patrons, de même que les images de *wireframe* le sont (comme le StyleGuide le propose, cf. §5.2.3). Il s'agirait alors de faire un pas dans l'étape de développement de l'application d'e-Procuration à concevoir.

Pour aller plus loin, nous pouvons nous inspirer des fonctionnalités offertes par les autres outils d'opérationnalisation de patrons. Ils fournissent dans la majorité des cas des moyens de visualiser la collection de patrons, certains représentent les relations entre eux.

- **Gestion d'une collection de patrons** : les utilisateurs expérimentés devraient être autorisés à ajouter facilement des enchaînements d'écrans qu'ils utilisent régulièrement et qui ne figurent pas encore dans le catalogue de patrons. L'utilisateur devrait pouvoir personnaliser son environnement de travail, comme il est proposé dans l'outil IdealXml (Montero et López-Jaquero 2006) ;
- **Partage d'une collection de patrons** : pour encourager la coopération entre membres d'une même communauté d'utilisateurs de patrons, les enchaînements d'écrans personnalisés ou les nouveaux patrons d'enchaînement d'écrans peuvent être mis à disposition d'autres utilisateurs, à la manière de l'outil CoPe *Collaborative Pattern Editor*[2] (Schobert et Schümmer 2006). Cet outil fournit également un moyen de visualiser les patrons et les relations entre eux.

---

[1] Langage de contraintes sur des objets

[2] Éditeur de patrons collaboratif

7. Conclusions

La modernisation des relations médiatisées par ordinateurs entre administrations et administrés est une vraie question d'actualité. Depuis quelques années, et notamment suite à la mise en place d'une politique volontariste par l'Administration Française pour simplifier les démarches administratives (décret du 25 mai 2001), des applications Web sont développées comme support aux procédures administratives (p.ex. télé déclaration d'impôt sur le revenu, demande d'intervention sur la voie publique, demande d'acte d'état civil). L'objectif de ces applications d'e-Procuration est double : du point de vue du citoyen, il s'agit de simplifier ses démarches auprès de l'administration, faciliter la communication avec les agents administratifs et rendre le processus administratif plus transparent ; du point de vue de l'administration, il s'agit de simplifier la gestion des dossiers. Parmi les objectifs annoncés, il y a la diminution des coûts, une meilleure circulation des informations, l'amélioration des conditions d'archivage, et la personnalisation des services.

Plusieurs travaux sur l'évaluation de l'utilisabilité et l'accessibilité des utilisateurs d'applications d'e-Gouvernement existent (Lili Wang, Bretschneider, et Gant 2005). Cependant, très peu d'études ont essayé de comprendre les difficultés rencontrées par les équipes de développement de ces applications. Il est important de remarquer que la prise en compte des besoins des concepteurs en termes de support à la communication et de guidage est un facteur important pour assurer la qualité des interfaces produites. Dans cette thèse nous nous sommes intéressés au processus de conception des applications d'e-Gouvernement. Après une analyse approfondie de la littérature et des observations menées sur le terrain, nous avons pu constater que le développement d'interfaces utilisateurs pour les applications d'e-Gouvernement est une activité très sensible et complexe. Ces applications peuvent être considérées sensibles à cause des données (d'ordre privé) qui sont manipulées et des conséquences des actes des utilisateurs (souvent avec des implications légales, p.ex. amende en cas de dépassement du délai pour réaliser sa déclaration d'impôts en ligne). Les applications d'e-Gouvernement peuvent être considérées comme complexes parce que leur développement requiert l'intervention de plusieurs corps de métiers qui doivent travailler conjointement, parce que ces applications doivent être mises à disposition d'un public très large, avec des interfaces soumises à des contraintes légales (p.ex. vie privée, accessibilité) ou parce que ces applications sont susceptibles d'évoluer en fonction de changements de la législation.

L'objectif principal des travaux réalisés est de proposer un cadre méthodologique pour guider les différents intervenants qui participent au processus de conception vers le développement d'applications utilisables. La méthode que nous proposons s'est appuyée sur la capitalisation de connaissances ergonomiques par le biais d'un catalogue de patrons de conception d'interfaces qui sont directement applicables à notre domaine d'étude, c'est-à-dire l'e-Procuration. Le travail qui a été réalisé autour de ce catalogue de patrons consiste à assembler des solutions existantes (dans la littérature et/ou sur le terrain) et à les organiser pour permettre une recherche clairvoyante parmi les bonnes pratiques de conception qu'ils proposent.

L'utilisation de patrons de conception d'interface a des implications multiples pour le domaine de l'e-Gouvernement. Tout d'abord, le catalogue de patrons de conception fournit un entrepôt unique de bonnes pratiques de conception qui est disponible à tout moment et pour tous les intervenants du processus de conception. L'impact immédiat de la mise en place de ce catalogue (telle qu'elle a été observée sur le terrain) est que les différents intervenants peuvent se servir de certains éléments du catalogue pour communiquer leurs idées et argumenter leurs choix de conception. Ce catalogue peut avoir un caractère officiel au sein d'une organisation et à ce titre contribuer à une démarche qualité des interfaces basée sur l'homogénéité des applications produites par une même organisation. Un apport majeur du catalogue concerne la présentation des informations à plusieurs niveaux d'abstraction allant du prototypage d'enchaînements

d'écrans aux consignes pour l'utilisation de composants de base, comme par exemple des menus.

La standardisation des interfaces basée sur un catalogue officiel pourrait être considérée comme un objectif à terme, pour assurer une cohérence transversale entre applications d'e-Gouvernement développées par les différents prestataires et sous-traitants des organismes publics. Il est important de faire remarquer que la démarche de standardisation au sens large est confrontée à la volonté d'adoption par d'autres organisations. En outre, compte tenu du caractère très orienté conception et implémentation du contenu des patrons, il faut mettre en place des mécanismes pour faire évoluer le catalogue de façon à prendre en compte de nouveaux types d'applications et les nouvelles technologies.

L'idée de présenter des solutions de conception d'interfaces par le biais d'un catalogue de patrons n'est pas nouvelle. Pourtant, l'originalité de ce travail repose sur le domaine d'application choisi (i.e. l'e-Gouvernement) et plus particulièrement sur la volonté de présenter les patrons de conception comme des éléments essentiels d'une méthode de conception des interfaces. La méthode proposée dans cette thèse est fondée sur des observations sur le terrain dans une démarche centrée utilisateur, qui en l'occurrence sont les conceptions des applications d'e-Procuration. Il est important de remarquer que cette méthode a été effectivement appliquée dans l'industrie sur des études de cas réelles. Après la mise en place de la méthode il nous a été possible de revenir pour évaluer son impact au sein de l'équipe de conception. À partir de cette évaluation nous avons pu valider nos hypothèses concernant l'apport de patrons de conceptions d'interface dans les étapes d'analyse fonctionnelle, le besoin de guidage pour la conception d'interfaces ergonomiques, l'importance de la capitalisation et mise à disposition de la connaissance en conception d'interfaces.

Cette étude nous a permis de comprendre les difficultés rencontrées par les équipes de conception pour développer des applications utilisables. Bien qu'il y ait un consensus sur l'importance de créer une culture de l'utilisabilité au sein des organisations, il est nécessaire que les efforts de diffusion de la connaissance soient soutenus par un cadre méthodologique qui puisse répondre aux besoins réels des différents intervenants dans le processus de conception. Au-delà du domaine de l'e-Gouvernement (et plus particulièrement des applications d'e-Procuration sur lesquelles nous nous sommes focalisés), la démarche que nous avons suivie pourrait être appliquée à d'autres domaines d'applications. Cependant, nous insistons sur l'importance des études de terrains pour prendre en compte les vrais besoins de différents intervenants dans le processus de conception. Donc, si la démarche globale de cette étude peut être considérée comme universelle, les contributions que nous apportons dans cette thèse sont volontairement limitées aux applications d'e-Gouvernement sur lesquelles nous avons eu l'occasion de valider nos hypothèses.

CONTRIBUTIONS

Ces travaux de thèse concernent l'introduction dans le milieu industriel d'un catalogue de patrons d'interface comme support à la conception d'applications d'e-Procuration ergonomiques. Nous avons validé en partie l'acceptabilité de cette méthode, en menant sur le terrain une étude évaluative sur une méthode similaire à la nôtre, déjà en place depuis plusieurs mois dans l'entreprise. Nous avons opérationnalisé notre méthode par un outil, eGovIPM, pour pallier aux problèmes de communication imprécise au sein d'une équipe pluridisciplinaire telle que celles que l'on rencontre en e-Gouvernement. Cet outil permet en effet, en plus du guidage de la personnalisation des patrons d'enchaînements d'écrans, de produire un support à la discussion avec le client d'un projet (squelette de l'application) automatiquement généré à partir d'un support qui, lui, peut être transmis à l'équipe de développement (modèle de navigation StateWebCharts). Une correspondance est ainsi établie entre deux artefacts à destination de deux catégories d'intervenants au processus de conception. Chacun d'eux dispose alors d'un artefact pouvant nourrir son activité et faire l'objet de discussions. Les deux parties traitent de deux vues

d’un même modèle de navigation, ce qui permet une communication indirecte mais sur la base de concepts communs. Les principales contributions de cette thèse sont détaillées ci-après.

Intégration de la méthode à un processus de conception basé sur le prototypage

Notre méthode de conception est inspirée d’une méthode de conception basée sur des patrons d’interfaces, observée dans l'entreprise SmalS. Les bonnes pratiques de conception traitant de problèmes de conception d’interfaces récurrents sont groupées dans un catalogue de patrons d'interfaces, disponible en ligne[1]. Notre méthode s’intègre à un processus de développement basé sur du prototypage moyenne fidélité. Pour accompagner le prototypage moyenne fidélité en phase d’analyse fonctionnelle, les patrons d’interface du catalogue sont associés à des prototypes moyenne fidélité génériques :

- *wireframes* pour les patrons de pages et de composants de base ;
- modèles de navigation StateWebCharts pour les patrons d’enchaînements d’écrans.

L’utilisateur de notre méthode adapte les prototypes génériques issus du catalogue de patrons au contexte particulier de l’application en cours de développement. Il est guidé en cela par des recommandations figurant dans le patron (p.ex. bons exemples d’implémentation du patron, cas d’utilisation) et, pour les patrons d’enchaînement d’écrans, par l’outil eGovIPM. Nous fournissons ainsi une aide à la spécification et à la production de supports à l’analyse fonctionnelle (*wireframes* et modèle de navigation) discutables avec le client, et l’équipe de développement.

Organisation de patrons d’interfaces pour l’e-Procuration

Les patrons sont organisés dans le catalogue selon leur niveau de granularité (enchaînements d'écrans, pages et composants de base) : le catalogue peut ainsi être consulté selon un ordre naturel de réflexion pour la conception, partant de problèmes de haut niveau d'abstraction vers des problèmes plus concrets.

Une proposition d’ontologie des concepts des patrons du catalogue complète cette organisation statique et offre des moyens de tisser des liens sémantiquement riches entre patrons. Elle fait pour cela figurer des informations n’apparaissant pas sur le catalogue pour ne pas surcharger la lecture des patrons, mais présentant un intérêt pour la recherche avancée de solutions adaptées à une situation de conception. Cette ontologie n’a pas, à l’heure actuelle, été intégrée au catalogue, mais elle a été développée séparément dans l’outil Protégé.

Évaluation de la faisabilité de la méthode

La faisabilité de notre méthode a été validée par l’évaluation d’une méthode similaire à la nôtre, basée sur des patrons d’interfaces pour l’e-Procuration, qui est d’ores et déjà déployée en entreprise : le StyleGuide (chapitre 5). Des observations de scénarios provoqués et des entretiens avec des utilisateurs du StyleGuide ont permis d’explorer des moyens d’améliorer l’acceptabilité de la méthode et de mieux l’adapter aux conditions de travail en entreprise.

Il a notamment été relevé que la communication au sujet du catalogue de patrons devait être soignée pour créer une communauté d’utilisateurs concernée par les informations capitalisées dans le catalogue. Nous recommandons de notifier régulièrement les utilisateurs du catalogue des changements de son contenu. L’aspect fortement évolutif du catalogue doit être présent dans l’esprit des utilisateurs, et il doit être clair que les utilisateurs sont le moteur principal de ces évolutions. Sans être responsables de la formalisation des patrons, ni de leur validité, ils sont les mieux placés, sur le terrain, pour suggérer des ajouts ou des modifications de patrons.

[1] http://fpontico.free.fr/ (dernier accès : 25 sept. 08)

Outillage de la méthode

Un outil d'aide à l'application de la méthode, eGovIPM, complète notre contribution à la conception d'interfaces en e-Gouvernement (chapitre 6). Il permet l'édition d'un modèle de navigation StateWebCharts qui schématise l'enchaînement d'écrans relatif à une application. L'utilisateur de l'outil est guidé dans cette activité :

- L'édition du modèle de navigation de son application est fait à partir d'un patron d'enchaînement d'écran générique, issu du catalogue ;
- Certaines activités d'édition sont interdites, lorsqu'elles enfreignent des contraintes relatives à l'enchaînement d'écran d'un patron (p.ex. dans un assistant à la réalisation d'une procédure, un récapitulatif doit figurer à la suite des étapes) ;

À partir du modèle de navigation édité, un squelette de l'application est généré : c'est un ensemble de pages XHTML, liées entre elles conformément au modèle de navigation édité. Le squelette généré peut être montré au client rapidement car il contient des *wireframes* de pages génériques si l'analyste n'a pas le temps d'éditer des *wireframes* personnalisés. Chaque page XHTML générée dispose du *wireframe* générique ou personnalisé qui lui correspond, d'une description saisie par le concepteur, et de liens hypertextes actifs, sous forme de liste, menant vers d'autres pages du squelette. Sans qu'il soit nécessaire de produire un prototype interactif en XHTML, un enchaînement des *wireframes* de page statiques peut ainsi être présenté au client. Pour ce qui concerne la communication avec l'équipe de développement, le squelette est un support pour l'expression précise des besoins collectés et spécifiés lors de l'analyse fonctionnelle.

PERSPECTIVES À COURT TERME

Ce travail de thèse porte sur la mise à disposition de connaissance ergonomique pour la conception, sans contribution sur la connaissance ergonomique elle-même. Toutefois, pour rendre notre méthode opérationnelle, il est nécessaire d'enrichir et de valider par inspection experte le contenu du catalogue de patrons, enchaînements d'écrans, pages et composants de base. L'ontologie des concepts des patrons doit être effectivement intégrée dans le catalogue. Une enquête préliminaire doit être réalisée auprès d'utilisateurs analystes pour connaître leur mode de raisonnement et en déduire des pistes de présentation de l'ontologie. La navigation au sein du catalogue pourra être rendue plus efficace par le biais de moyens de visualisation adaptés (p.ex. réseau en étoile ou arbre) ou la mise à disposition de fonctionnalités de personnalisation. L'ontologie doit prendre en compte des attributs tels que le niveau d'accessibilité d'un patron, ou les dispositifs physiques pour lesquels il est le plus adapté (p.ex. navigateur Web ou PDA). L'utilisateur du catalogue pourrait ainsi préciser ses contraintes pour naviguer dans le catalogue, par exemple « Accessibilité de niveau Bronze exigée (label AccessiWeb)[1] », ou encore « Lisibilité depuis un PDA exigée ».

Des contraintes plus fines peuvent être définies dans l'outil, et notamment concernant des contraintes d'accessibilité ou d'utilisabilité sur des dispositifs physiques offrant des modes d'interaction réduits (p.ex. téléphone portable). Des contraintes de respect de recommandations ergonomiques enrichiraient encore le guidage de l'édition de la structure hypertexte de l'application en cours de conception (p.ex. « Toute page doit disposer d'un lien entrant et d'un lien sortant au moins »). Pour plus de souplesse, et pour favoriser l'évolution de l'outil, les contraintes d'édition des modèles d'enchaînements d'écrans doivent être détachées du code de l'outil, et définies dans un fichier à part. Les extraire du code Java permettrait de les éditer et de les personnaliser plus facilement ; formaliser leur définition avec un format de type XML permettrait de les exploiter de façon plus systématique.

[1] http://www.accessiweb.org/fr/Label_Accessibilite/ (dernier accès : 25 sept. 08)

PERSPECTIVES À LONG TERME

Plusieurs domaines d'applications hypermédia ont leur propre ensemble de patrons d'interfaces (p.ex. e-Learning (Derntl et Motschnig-Pitrik 2004)) : notre méthodologie ajouterait un cadre formel à leur utilisation, ainsi que la spécification de la structure hypertexte qui n'apparaît pas dans ces patrons. Pour les domaines ne disposant pas de patrons, notre méthodologie de collecte d'éléments récurrents par analyse de l'existant peut être appliquée, pour constituer une base de patrons qui émergent de leur domaine d'application. Il est également nécessaire, comme nous l'avons fait, d'analyser le travail des utilisateurs des patrons pour intégrer la méthode au processus de conception, et la rendre acceptable.

Notre méthode pourrait être étendue à d'autres activités de conception, en enrichissant le catalogue avec des patrons ne concernant pas l'interface mais d'autres facettes de l'application. Des patrons de composants de base avec des fragments de code pourraient être ajoutés tout d'abord, de la manière dont a procédé SmalS (cf. §5.2.3), fournissant un support au prototypage interactif. Des patrons de modélisation de procédure existent déjà pour l'e-Gouvernement (Verginadis, Gouscos, et Mentzas 2004; Momotko et coll. 2007; Ouchetto, Fredj, et Roudiès 2007). Les formaliser et les intégrer au catalogue voire à un outil d'édition du type d'eGovIPM permettrait de guider la modélisation de la procédure, étape capitale et sujette à évolutions de la conception d'une application d'e-Procuration. Des patrons de modélisation de tâches, de domaine, d'organisation pourraient intervenir également pour les procédures complexes. Au niveau de l'outil, des réflexions pourraient être menées sur les moyens d'articuler et rendre cohérents ces modèles, à la manière de l'outil PIM *Patterns In Modelling*[1] (Radeke et coll. 2006) qui autorise la combinaison de patrons d'interfaces, chacun étant régi par un modèle. Ces perspectives de travail permettraient de guider plus encore les concepteurs dans la synchronisation des multiples contraintes qui rythment la conception d'une application d'e-Gouvernement.

[1] Patrons dans la modélisation

Bibliographie

Accenture. 2005. *Leadership in Customer Service: New Expectations, New Experiences*. The Government Executive Series. Avril 2005. http://www.accenture.com/NR/rdonlyres/F9C0479F-5139-49E3-A71D-5895AA145C6A/0/leadership_cust.pdf

AGIMO. 2006. *Australians' use of and satisfaction with e-Government services*. Étude. Australie Juillet 2006. http://www.agimo.gov.au/__data/assets/pdf_file/0007/51496/2006_Measurementreport_final.pdf

Agostini, Pietro Luca, et Raffaella Naggi. 2007. A critical analysis of the concepts of accessibility and usability of e-Government web sites in Italian legislation. A comparison with the German approach. Dans *EGOV (ongoing research, project contributions and workshops)*, 183-193. Regensburg, Allemagne: Trauner Verlag, 3-7 Septembre 2007.

Alani, Harith. 2003. TGVizTab: An Ontology Visualisation Extension for Protégé. Dans *Workshop on Visualization Information in Knowledge Engineering (at K-Cap)*. Sanibel Island, Florida, USA, 25-26 Octobre 2003.

Alexander, Christopher, Sara Ishikawa, Murray Silverstein, et coll. 1977. *A pattern language*. New York, USA: Oxford University Press.

Axelsson, Karin, et Ulf Melin. 2007. Talking to, not about, citizens – Experiences of focus groups in public e-service development. Dans *EGOV*, 179-190. Regensburg, Allemagne: Springer Berlin / Heidelberg, 3-7 Septembre 2007.

Axelsson, Karin, et Stefan Ventura. 2007. Reaching Communication Quality in Public E-Forms – A Communicative Perspective on E-Form Design. Dans *EGOV*, 342-353. Regensburg, Allemagne: Springer Berlin / Heidelberg, 3-7 Septembre 2007.

Barna, Peter, Flavius Frasincar, et Geert-Jan Houben. 2006. A workflow-driven design of web information systems. Dans *6th international conference on Web engineering*, 321-328. Palo Alto, Californie, USA: ACM, 12-14 Juillet 2006.

Barsalo, René. 2004. Le dé « clic » générationnel. Conférence présentée à la Journée de l'informatique du Québec, 10 Novembre 2004, Québec, Canada. http://www.opossum.ca/jiq2004/Barsalo.ppt

Barta, Robert A., et Markus W. Schranz. 1998. JESSICA: an object-oriented hypermedia publishing processor. *Computer Networks and ISDN Systems* 30, no. 1-7: 281-290.

Bastide, Rémi. 1992. Objets Coopératifs : Un formalisme pour la modélisation des systèmes concurrents. Thèse de Doctorat de l'Université Toulouse 1, France.

Bastien, J. M. Christian, et Dominique L. Scapin. 1995. Evaluating a user interface with ergonomic criteria. *Int. J. Hum.-Comput. Interact.* 7, no. 2: 105-121.

Bastien, J. M. Christian, et Dominique Louis Scapin. 1993. *Ergonomic criteria for the evaluation of human-computer interfaces*. Rapport technique. Rocquencourt, France: INRIA, Juin.

Beirekdar, Abdo, Jean Vanderdonckt, et Monique Noirhomme-Fraiture. 2002. A Framework and a Language for Usability Automatic Evaluation of Web Sites by Static Analysis of HTML Source Code. Dans *CADUI*. Valenciennes, France: Kluwer Academics Pub., 15-17 Mai 2002.

Berners-Lee, Tim, James Hendler, et Ora Lassila. 2001. The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities.

Scientific American Magazine, Mai 2001. http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

Beyer, Hugh, et Karen Holtzblatt. 1997. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann Publishers.

Bicharra Garcia, Ana Cristina, Cristiano Maciel, et Fernando Bicharra Pinto. 2005. A Quality Inspection Method to Evaluate E-Government Sites. Dans *EGOV*, 198-209. Copenhague, Danemark: Springer Berlin / Heidelberg, 22-26 Août 2005.

Birou, Alain. 1966. Modèle. Dans *Vocabulaire pratique des sciences sociales*.

Boehm, Barry. 1986. A spiral model of software development and enhancement. *SIGSOFT Softw. Eng. Notes* 11, no. 4: 14-24.

Boersma, Peter. 2005. Introducing User-Centered Design to an E-Government Software Development Company. *ASIS&T Bulletin*, Mars.

Borchers, Jan O. 1997. WorldBeat: Designing a Baton-Based Interface for an Interactive Music Exhibit. Dans *CHI*, 131-138, Atlanta, Georgia, USA: ACM/Addison-Wesley, 22-27 Mars 1997.

---. 2000. A pattern approach to interaction design. Dans *Designing interactive systems*, 369-378. New York City, USA: ACM Press.

Brain, David, Philip Seltsikas, et Deemple Tailor. 2005. Process modelling notations for eGovernment: an assessment of modelling notations for identity management. Dans *eIntegration in Action*. Bled, Slovénie, 6-8 Juin 2005.

Brajnik, Giorgio. 2000. Automatic web usability evaluation: what needs to be done? Dans *6th Human Factors and the Web Conference*. Austin, Texas, USA, 19 Juin 2000.

Brams, G. W. 1983. *Réseaux de Petri : Théorie et pratique*. Masson.

Brown, William J., Raphael C. Malveau, et Thomas J. Mowbray. 1998. *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*. 1er éd. Wiley, Mars 1998.

BUCOM. 2007. Manuel Prima-Web. Août 2007. https://www.socialsecurity.be/site_fr/civilservant/Applics/primaweb/documents/pdf/utilisateur_primaweb_F.pdf

Caelen, Jean. 2004. Conception participative des objets interactifs : principes, méthodes et instrumentation présenté à l'École d'été TIC et société, CNRS, Carry le Rouet. http://www-clips.imag.fr/geod/User/jean.caelen/Publis_fichiers/CaelenCarry2004.pdf

CapGemini Netherlands, et TNO. 2004. *Does e-Government Pay Off?* Europe: epractice.eu, 22 Novembre 2004. http://www.epractice.eu/document/2951

CapGemini. 2006. *Online Availability of Public Services: How Is Europe Progressing?* Juin 2006. http://ec.europa.eu/information_society/eeurope/i2010/docs/benchmarking/online_availability_2006.pdf

Card, Stuart K., Allen Newell, et Thomas P. Moran. 1983. *The Psychology of Human-Computer Interaction*. Mahwah, New Jersey, USA: Lawrence Erlbaum Associates, Inc.

Carlsen, Steinar. 1998. Action Port Model: a mixed paradigm conceptual workflow modeling language. Dans *3rd IFCIS International Conference on Cooperative Information Systems*. New York City, New York, USA: IEEE-CS Press, 20-22 Août 1998.

Ceri, Stefano, Pierro Fratemali, et Aldo Bongio. 2000. Web Modeling Language (WebML): a Modeling Language for Designing Web Sites. Dans *WWW9*. Amsterdam, Pays-Bas: Elsevier Science B.V., 15-19 Mai 2000.

Clement, Andrew, et Peter Van den Besselaar. 1993. A retrospective look at PD projects. *Communications of the ACM* 36, no. 6 (Juin 1993): 29-37.

CNIL. 2004. *Loi du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés*. 6 Août 2004.

Codd, E. F. 1970. A relational model of data for large shared data banks. *Communications of the ACM* 13, no. 6: 377-387.

Collins, Dave. 1994. *Designing Object-Oriented User Interfaces*. Benjamin-Cummings Publishing Co., Inc.

Cooper, Alan. 1999. *The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Indianapolis, USA: Sams. http://www.infotoday.com/Online/jul03/head.shtml#essential

Coram, Todd, et Jim Lee. 2002. Experiences - A pattern language for user interface design. *http://www.maplefish.com/todd/papers/Experiences.html*

Coutaz, Joelle. 1988. Interface Homme-Ordinateur : Composition et réalisation. Thèse de doctorat de l'Université Joseph Fourier, Grenoble, France.

Cowan, Donald D., et Carlos J. P. Lucena. 1995. Abstract Data Views: An Interface Specification Concept to Enhance Design for Reuse. *IEEE Transactions on software engineering* 21, no. 3: 229-243.

Curtis, Bill, et Bill Hefley. 1994. A wimp no more: the maturing of user interface engineering. *interactions* 1, no. 1: 22-34.

Dearden, Andy, et Janet Finlay. 2006. Pattern languages in HCI: a critical review. *Human Computer Interaction* 21, no. 1: 49-102.

De Troyer, Olga, et Sven Casteleyn. 2003. Exploiting link types during the conceptual design of websites. *International journal of Web engineering and technology* 1, no. 1: 17-40.

Derntl, Michael, et Renate Motschnig-Pitrik. 2004. A Pattern Approach to Person-Centered e-Learning Based on Theory-Guided Action Research. Dans *Networked Learning Conference 2004*. Lancaster, UK, 5-7 Avril 2004.

Distante, Damiano, Gustavo Rossi, Gerardo Canfora, et Scott Tilley. 2007. A comprehensive design model for integrating business processes in web applications. *International Journal of Web Engineering and Technology* 3, no. 1: 43-72.

DmfA. 2004. Manuel de la nouvelle déclaration ONSS. 16 Janvier 2004. https://www.socialsecurity.be/site_fr/Applics/dmfa/documents/pdf/manual_dmfa_F.pdf

Dumas, Philippe, et Gilles Charbonnel. 1990. *La méthode OSSAD : pour maîtriser les technologies de l'information*. Vol. 1. Les Éditions d'Organisation.

Eclipse. 2008a. Eclipse.org. http://www.eclipse.org/

---. 2008b. Eclipse Graphical Editing Framework (GEF). http://www.eclipse.org/gef/

Fincher, Sally, et Peter Windsor. 2000. Why patterns are not enough: some suggestions concerning an organizing principle for patterns of UI design. Dans *Workshop Pattern languages for interaction design: building momentum (at CHI2000)*. La Hague, Pays-Bas: ACM, 31 Mars – 5 Avril 2000.

Flanders, Vincent. 2003. Fixing Your Web Site using Dr. HTML. *Dr. HTML*. http://www.fixingyourwebsite.com/drhtml.html

Fleming, Jennifer. 1998. *Web navigation: designing the user experience*. Sebastopol, California, USA: O'Reilly & Associates, Inc.

Floyd, Christine, Wolf-Michael Mehl, Fanny-Michaela Resin, Gerhard Schmidt, et Gregor Wolf. 1989. Out of Scandinavia: Alternative Approaches to Software Design and System Development. *Human-Computer Interaction* 4, no. 4: 253.

Følstad, Asbjørn, Håvard D. Jørgensen, et John Krogstie. 2004. User involvement in e-government development projects. Dans *Proceedings of the third Nordic conference on Human-computer interaction*, 217-224. Tampere, Finlande: ACM, 23-27 Octobre 2004.

Fraser, John, Nick Adams, Ann Macintosh, et coll. 2003. Knowledge management applied to e-Government services: the use of an ontology. Dans *KMGov*, 116-126. Rhodes Island, Grèce: Springer Berlin / Heidelberg, 26 Mai 2003.

Gaffney, Gerry. 1998. Web site evaluation checklist v1.1. *Information & Design pty ltd*. http://www.infodesign.com.au/ftp/WebCheck.pdf

Gamma, Erich, Richard Helm, Ralph Johnson, et John M. Vlissides. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, Novembre 1994.

Genitech. 2007. e-Citiz, e-Service Factory. http://www.ecitiz.com/

Glassey, Olivier. 2005. Knowledge Component-based Architecture for Process Modelling. Dans *eGOV INTEROP'05*, 23-24. Genève, Suisse, Février 2005.

Glassey, Olivier, et Jean-Loup Chappelet. 2002. *Comparaison de trois techniques de modélisation de processus: ADONIS, OSSAD et UML*. Rapport technique. Lausanne, Suisse: IDHEAP.

GlassFish. 2008. JAXB Reference Implementation. https://jaxb.dev.java.net/

Gruber, Thomas R., et Daniel M. Russell. 1990. *Design Knowledge and Design Rationale: A Framework for Representation, Capture, and Use*. Rapport technique. Knowledge Systems Laboratory, Stanford University. http://citeseer.ist.psu.edu/article/gruber91design.html

Hackos, Joann T., et Janice C. Redish. 1998. *User and task analysis for interface design*. New York, USA: Wiley.

Halstead-Nussloch, Richard, Doris Konneh, et Robert Woodruff. 2003. Communities of design practice in electronic government. Dans *CHI '03 extended abstracts on Human factors in computing systems*, 744-745. Ft. Lauderdale, Florida, USA: ACM, 5-10 Avril 2003.

Handzic, Meliha. 2004. The Role of Knowledge Mapping in Electronic Government. Dans *KMGov*, 9-17. Krems, Autriche: Springer Berlin, 17 Mai 2004.

Harel, David. 1987. Statecharts: A visual Formalism for complex systems. *Science of Computer Programming* 8, no. 3: 231-274.

Harper, Ben, Laura Slaughter, et Kent Norman. 1997. Questionnaire administration via the WWW: A validation & reliability study for a user satisfaction questionnaire. Dans *Web-Net*. Toronto, Canada: AACE, 1-5 Novembre 1997.

Hartson, H. Rex, et José Castillo. 1998. Remote evaluation for post-deployment usability improvement. Dans *AVI*. L'Aquila, Italie: ACM Press, 24-27 Mai 1998.

Hartson, H. Rex, José C. Castillo, John Kelso, et Wayne C. Neale. 1996. Remote evaluation: The Network as an extension of the usability laboratory. Dans *CHI*, 228-235. Vancouver, British Columbia, Canada: ACM, 13-18 Avril 1996.

Henninger, Scott, et Padmapriya Ashokkumar. 2005. An ontology-based infrastructure for usability patterns. Dans *Workshop on Semantic web enabled software engineering (at ISWC)*. Galway, Irlande, 6-10 Novembre 2005.

Hix, Deborah, et Rex Hartson. 1993. *Developing user interfaces: ensuring usability through product and process*. New York, USA: John Wiley & Sons.

Hoogendoorn, Mark, Catholijn M. Jonker, Martijn C. Schut, et Jan Treur. 2007. Modeling centralized organization of organizational change. *Computational & Mathematical Organization Theory* 13, no. 2 (11 Juin 2007): 147-184.

Huber, Peter, Kurt Jensen, et Robert M. Shapiro. 1990. Hierarchies in Coloured Petri Nets. Dans *Advances in Petri Nets*, 313-341. Paris, France: Springer Verlag.

Isakowitz, Tomás, Edward A. Stohr, et P. Balasubramanian. 1995. RMM: a methodology for structured hypermedia design. *Commun. ACM* 38, no. 8: 34-44.

Ivory, Melody Y. 2003. *Automated web site evaluation: researchers' and practitioners' perspectives*. Dordrecht, Pays-Bas: Kluwer Academic Publishers.

Ivory, Melody Y., et Marti A. Hearst. 2002. Improving web site design. *IEEE Internet Computing* 6, no. 2: 56-63.

Jacobs, Ian. 1992. *The Centaur 2.0 Manual*. Research report. Sophia Antipolis, France: INRIA.

Johnson, Peter. 1992. *Human computer interaction : psychology, task analysis and software engineering*. Berkshire, UK: McGRAW-HILL Book Company Europe.

Johnson, Peter, Stephanie Wilson, Panos Markopoulos, et James Pycock. 1993. ADEPT - Advanced Design Environment for Prototyping with Task Models. Dans *INTERCHI'93*. Amsterdam, Pays-Bas: ACM Press, 24-29 Avril 1993.

Joshi, Suneela R., et William W. McMillan. 1996. Case based reasoning approach to creating user interface components. Dans *Conference companion on Human factors in computing systems: common ground*, 81-82. Vancouver, British Columbia, Canada: ACM, 13-18 Avril 1996.

Katifori, Akrivi, Constantin Halatsis, George Lepouras, Costas Vassilakis, et Eugenia Giannopoulou. 2007. Ontology visualization methods - a survey. *ACM Comput. Surv.* 39, no. 4: 10.

Kavadias, Gregory, et Efthimos Tambouris. 2003. GovML: a markup language for describing public services and life events. Dans *4th working conference on Knowledge Management in Electronic Government*. Rhodes Island, Grèce: Springer Berlin / Heidelberg, 26-28 Mai 2003.

Kay, Michael. 2008. The SAXON XSLT and XQuery Processor. http://saxon.sourceforge.net/

Kirakowski, Jurek, et Nigel Claridge. 1998. Human Centered Measures of Success in Web Site Design. Dans *4th Conference on Human Factors & the Web*. Basking Ridge, NJ, USA, 5 Juin 2008.

Knapp, Alexander, Nora Koch, Flavia Moser, et Gefei Zhang. 2003. ArgoUWE: A Case Tool for Web Applications. Dans *Workshop on Engineering Methods to Support Information Systems Evolution*. Geneva, Suisse, 5 Septembre 2003.

Koch, Nora, Hubert Baumeister, Rolf Hennicker, et Luis Mandel. 2000. Extending UML to Model Navigation and Presentation in Web Applications. Dans *Workshop on Modelling Web Applications (at UML)*. York, UK, Octobre 2000.

Koch, Nora, Andreas Kraus, Cristina Cachero, et Santiago Melia. 2003. Modeling Web Business Processes with OO-H and UWE. Dans *IWWOST'03 (3rd International Workshop on Web Oriented Software Technology)*. Oviedo, Espagne, 15 Juillet 2003.

Kolski, Christophe. 1998. A "call for answers" around the proposition of an HCI-enriched model. *SIGSOFT Softw. Eng. Notes* 23, no. 3: 93-96.

Kotzé, Paula, Karen Renaud, et Judy van Biljon. 2008. Don't do this - Pitfalls in using anti-patterns in teaching human-computer interaction principles. *Computers & Education* 50, no. 3 (Avril 2008): 979-1008.

Kruchten, Philippe. 2000. *The Rational Unified Process: An Introduction*. 2 éd. Addison-Wesley Professional, 14 Mars 2000.

Kunstelj, Mateja, Tina Jukic, et Mirko Vintar. 2007. Analysing the demand side of e-Government: what can we learn from Slovenian users? Dans *EGOV*, 305-317. Regensburg, Germany: Springer Verlag, 3-7 Septembre 2007.

Laakso, Sari A. 2003. User interface design patterns. *http://www.cs.helsinki.fi/u/salaakso/patterns/*

Lacaze, Xavier. 2005. La conception rationalisée pour les systèmes interactifs. Thèse de doctorat de l'Université Toulouse 1, France, Juin 2005. http://liihs.irit.fr/lacaze/divers/These_Xavier_Lacaze.pdf

Ladd, Ivor, et Dennis Tsichritzis. 1980. An Office Form Flow Model. Dans *AFIPS National Computer Conference*, 533-539. Anaheim, California, USA: AFIPS Press, Mai 1980.

Lee, Heeseok, Choongseok Lee, et Cheonsoo Yoo. 1999. A scenario-based object-oriented hypermedia design methodology. *Information & Management* 36, no. 3 (Septembre 1999): 121-138.

Lepreux, Sophie, Mourad Abed, et Christophe Kolski. 2003. A human-centred methodology applied to decision support system design and evaluation in a railway network context. *Cognition, Technology & Work* 5, no. 4 (1er Décembre 2003): 248-271.

Lewis, Clayton, Peter G. Polson, Cathleen Wharton, et John Rieman. 1990. Testing a walk-through methodology for theory-based design of walk-up-and-use interfaces. Dans *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, 235-242. Seattle, Washington, USA: ACM, 1-5 Avril 1990.

Loukis, Euripidis, et Spyros Kokolakis. 2003. Computer supported collaboration in the Public Sector: the ICTE-PAN Project. Dans *EGOV 2003*, 181-186. Prague, République Tchèque: Springer Berlin / Heidelberg, 1er Septembre 2003.

Lucquiaud, Vincent. 2005. Sémantique et outil pour la modélisation des tâches utilisateur : N-MDA. Thèse de doctorat de l'Université de Poitiers, France, Décembre 2005.

Maguire, M.C. 1999. A review of user-interface design guidelines for public information kiosk systems. *Internation Journal of Human-Computer Studies* 50: 263-286.

Mahemoff, Michael J., et Lorraine J. Johnston. 1998. Principles for a Usability-Oriented Pattern Language. Dans *Australasian Conference on Computer Human Interaction*, 132. Adelaide, Australie: IEEE Computer Society, 30 Novembre – 4 Décembre 1998.

Malone, Eric, Matt Leacock, et Chanel Wheeler. 2005. Implementing a Pattern Library in the Real World: A Yahoo! Case Study. Dans *ASIS&T*. Montréal, Canada, 28 Octobre – 2 Novembre 2005.

Mariage, Céline, Jean Vanderdonckt, Abdo Beirekdar, et Monique Noirhomme. 2004. DESTINE: outil d'aide à l'évaluation de l'ergonomie des sites web. Dans *Proceedings of the 16th conference on Association Francophone d'Interaction Homme-Machine*, 117-124. Namur, Belgique: ACM, 30 Août – 3 Septembre 2004.

Mariage, Céline, Jean Vanderdonckt, et Costin Pribeanu. 2005. State of the Art of Web Usability Guidelines. Dans *The Handbook of Human Factors in Web Design*, Chapitre 41.

Martin, David, Mark Rouncefield, et Ian Sommerville. 2002. Applying patterns of cooperative interaction to work (re)design: e-government and planning. Dans *Human factors in Computing Systems*, 235-242. Minneapolis, Minnesota, USA: ACM, 20-25 Avril 2002.

McConnell, Steve M. 1996. *Rapid Development: Taming Wild Software Schedules*. Microsoft Press, Août 1996.

McCurdy, Michael, Christopher Connors, Guy Pyrzak, Bob Kanefsky, et Alonso Vera. 2006. Breaking the fidelity barrier: an examination of our current characterization of prototypes and an example of a mixed-fidelity success. Dans *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 1233-1242. Montréal, Québec, Canada: ACM, 24-27 Avril 2006.

McDermid, John A., et Knut Ripken. 1984. *Life Cycle Support in the Ada Environment*. New York, NY, USA: Cambridge University Press, Février 1984.

Meyer, Bonnie J.F., et G. Elizabeth Rice. 1992. Prose Processing in Adulthood: The Text, the Reader, and the Task. Dans *Everyday Cognition in Adulthood and Late Life*, 157-194. New York, NY, USA: Leonard W. Poon and B.A. Wilson.

Momotko, Mariusz, Wojciech Izdebski, Efthimios Tambouris, Konstantinos Tarabanis, et Mirko Vintar. 2007. An architecture of active life event portals: generic workflow approach. Dans *EGOV*, 104-115. Regensburg, Allemagne: Springer Berlin / Heidelberg, 3-7 Septembre 2007.

Montero, Francisco, Maria Lozano, Pascual Gonzalez, et Isidros Ramos. 2002. Designing websites by using patterns. Dans *SugarLoafPLoP*, 209-224. Itaipava, Brésil: State University of São Paulo, 5-7 Août 2002.

Montero, Francisco, et Víctor López-Jaquero. 2006. IdealXml: An Interaction Design Tool. Dans *Computer-Aided Design Of User Interfaces*, 245-252. Bucharest, Roumanie: Springer-Verlag, 5-8 Juin 2006.

Morgan, David L. 1997. *The Focus Group Guidebook*. 1er éd. Sage Publications, Inc, 24 Juillet 1997.

Navarre, David, Philippe Palanque, Rémi Bastide, Fabio Paterno, et Carmen Santoro. 2001. A tool suite for integrating task and system models through scenarios. Dans *DSV-IS*. Glasgow, Écosse, UK: LNCS 2220 Springer, 13-15 Juin 2001.

Nielsen, Jakob. 1990. Improving a human-computer dialogue: What designers know about traditional interface design. *Communications of the ACM* 33, no. 3 (Mars 1990).

---. 1993. *Usability engineering*. London, UK: Academic Press Limited.

---. 1994. Heuristic evaluation. Dans *Usability inspection methods*, 25-62. New York, NY, USA: John Wiley & Sons Inc.

---. 2005. Heuristics for User Interface Design. *useit.com*. http://www.useit.com/papers/heuristic/heuristic_list.html

---. 2006. Alertbox. *http://www.useit.com/alertbox/*

Nielsen, Jakob, et Thomas K. Landauer. 1993. A mathematical model of the finding of usability problems. Dans *INTERCHI'93*, 206-213. Amsterdam, Pays-Bas: ACM, 24-29 Avril 1993.

Niman, Bruno von, Alejandro Rodriguez-Ascaso, Steve Brown, et Torbjorn Sund. 2007. User experience design guidelines for telecare (e-health) services. *interactions* 14, no. 5: 36-40.

Noble, James. 1998. Classifying relationships between object-oriented design patterns. Dans *Australian Software Engineering Conference*, 98-109. Adelaide, Australie: IEEE Computer Society, 9-13 Novembre 1998.

Norguet, Jean-Pierre, Quentin Limbourg, et Paul Stijfhals. 2007. Web metrics. Session d'information externe SmalS Bruxelles, Belgique. http://www.smals.be/site_fr/components/file/WebMetrics.pdf

Olsen, Henrik. 2002. Results from a survey of web prototyping tools usage. *guuui.com*, 30 Septembre 2002. http://www.guuui.com/issues/01_03_02.php

Olson, Ryan, Jessica Verley, Lindsay Santos, et Coresta Salas. 2004. What We Teach Students About the Hawthorne Studies: A Review of Content Within a Sample of Introductory I-O and OB Textbooks. *The Industrial-Organizational Psychologist* 41, no. 3: 23-39.

OMG. 2007. OMG OCL specification page. http://www.omg.org/technology/documents/modeling_spec_catalog.htm#OCL

Oostveen, Anne-Marie, et Peter Van den Besselaar. 2004. From small scale to large scale user participation: a case study of participatory design in e-government systems. Dans *8th conference on Participatory design*, 1:173-182. Toronto, Ontario, Canada: ACM, 27-31 Juillet 2004.

Ouchetto, Hassania, Mounia Fredj, et Ounsa Roudiès. 2007. Des patterns processus pour l'e-gouvernement. *e-TI - la revue électronique des technologies d'information*, no. 4 (30 Juin 2007). http://www.revue-eti.net/document.php?id=1467

Paterno, Fabio, Carmen Santoro, et Sophie Tahmassebi. 1998. Formal Models for Cooperative Tasks - Concepts and an Application for En-Route Air Traffic Control. Dans *DSV-IS*. Abingdon, UK: Springer, 3-5 Juin 1998.

Pearce, Jon M., John Murphy, et David Patman. 2006. Using a scenario-planning tool to support an engaging online user experience. Dans *CHISIG*, 167-174. Sydney, Australie: ACM, 20-24 Novembre 2006.

Perzel, Kimberly, et David Kane. 1999. Usability Patterns for Applications on the World Wide Web. Dans *PloP*. Urbana, IL, USA, 15-18 Août 1999.

Pierret-Golbreich, Christine, Isabelle Delouis, et Dominique Scapin. 1989. *Un outil d'acquisition et de représentation des tâches orienté objet*. Rapport d'activités. Le Chesnay, France: INRIA Rocquencourt. ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-1063.pdf

Pinker, Steven. 1999. *How the Mind Works*. 1er éd. W. W. Norton & Company, Janvier 1.

Pontico, Florence. 2008. Patrons eGov. Catalogue de patrons d'interface. *Patrons d'interfaces pour l'e-Gouvernement*. http://fpontico.free.fr/

Pontico, Florence, Christelle Farenc, et Marco Winckler. 2006. Model-based support for specifying eService eGovernment applications. Dans *TAMODIA*. Hasselt, Belgique: LNCS 4385 Springer, 23-24 Octobre 2006.

Pontico, Florence, Marco Winckler, et Quentin Limbourg. 2007a. Organizing user interface patterns for e-Government applications. Dans *EIS-HCSE-DSVIS*. Salamanca, Espagne: LNCS Springer, 22-24 Mars 2007.

---. 2007b. Towards a universal catalogue of User Interface patterns for e-Government Web sites. Dans *EGOV*, 227-232. Regensburg, Allemagne: Trauner Druck, 3-7 Septembre 2007.

Prinz, Wolfgang. 1994. Object-Oriented Organization Modeling for the Support of CSCW. Dans *HICSS*, 797-806. Hawaii, USA: IEEE CS Press, 4-7 Janvier 1994.

Pruitt, John, et Jonathan Grudin. 2003. Personas: practice and theory. Dans *Proceedings of the 2003 conference on Designing for user experiences*, 1-15. San Francisco, California: ACM.

Radeke, Frank, Peter Forbrig, Ahmed Seffah, et Daniel Sinnig. 2006. PIM Tool: Support for Pattern-Driven and Model-Based UI Development. Dans *TAMODIA*, 82-96. Hasselt, Belgique: LNCS 4385 Springer, 23-24 Octobre 2006.

Reenskaug, Trygve. 1979. *THING-MODEL-VIEW-EDITOR an Example from a planning system*. Note technique. Xerox PARC. http://heim.ifi.uio.no/~trygver/1979/mvc-1/1979-05-MVC.pdf

Reeves, Leah M., Jennifer Lai, James A. Larson, et coll. 2004. Guidelines for multimodal user interface design. *Commun. ACM* 47, no. 1: 57-59.

Reuchlin, Maurice. 1992. *Psychologie*. 15 éd. Presses Universitaires de France - PUF.

Ronkko, Kari. 2005. An Empirical Study Demonstrating How Different Design Constraints, Project Organization and Contexts Limited the Utility of Personas. Dans *HICSS*, 220.1. IEEE Computer Society, 3-6 Janvier 2005.

Royce, W. W. 1970. Managing the development of large software systems: concepts and techniques. Dans *IEEE WESCON 08*, 1-9. Monterey, California, USA.

Rumbaugh, James, Grady Booch, et Ivar Jacobson. 1999. *The Unified Modelling Language Reference Manual*. Reading, Massachusetts, USA: Addison Wesley Longmann, Inc.

Scapin, Dominique, Jean Vanderdonckt, Christelle Farenc, et coll. 2000. Transferring knowledge of user interfaces guidelines to the Web. Dans *International Workshop on Tools for Working with Guidelines TFWWG*, 293-304. Biarritz, France: Springer-Verlag, Londres, 7-8 Octobre 2000.

Scapin, Dominique Louis. 1990. Des critères ergonomiques pour l'évaluation et la conception d'interfaces utilisateurs. Dans *XXVI congrès de la SELF*. Montréal, Canada, 3 Octobre 1990.

Scapin, Dominique Louis, Corinne Leulier, Jean Vanderdonckt, et coll. 2000. A framework for organizing web usability guidelines. Dans *6th conference on human factors and the web*. Austin, Texas, USA: Ph. Kortum & E. Kudzinger (éds.), 19 Juin 2000.

Scharl, Arno. 1999. A Conceptual, User-Centric Approach to Modeling Web Information Systems. Dans *AusWeb*, 33-49. Ballina, Australie, 18-20 Avril 1999.

Schobert, Wolfram, et Till Schümmer. 2006. Supporting Pattern Language Visualization with CoPE. Dans *EuroPLoP*. Irsee, Allemagne: Universitaetsverlag Konstanz, 5-9 Juillet 2006.

Schwabe, Daniel, et Gustavo Rossi. 1998. Developing hypermedia applications using OOHDM. Dans *Workshop on Hypermedia Development*. Pittsburgh, PA, USA: Peter Brusilovsky, Paul De Bra (éds.), 20-24 Juin 1998.

Section508.gov. 2008. *Section 508 Standards*. http://www.section508.gov/index.cfm?FuseAction=Content&ID=12

Shiu, Simon, et Sankar K. Pal. 2004. *Foundations of Soft Case-Based Reasoning*. 1er éd. Wiley-Interscience, 18 Mars 2004.

Stanford Center for Biomedical Informatics Research. 2008. The Protégé Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu/

Storey, Margaret-Anne, Mark Musen, John Silva, et coll. 2001. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protege. Dans *Workshop on Interactive Tools for Knowledge Capture (at K-CAP)*. Victoria, B.C., Canada, 20 Octobre 2001.

Storrs, Graham. 1995. The notion of task in Human Computer Interaction. Dans *BCS HCI*, 357-365. Huddersfield, UK: Cambridge University Press, Août 1995.

Sunagawa, Eiichi, Kouji Kozaki, Yoshinobu Kitamura, et Riichiro Mizoguchi. 2006. Role Organization Model in Hozo. Dans *Managing Knowledge in a World of Networks*, 67-81. Podebrady, République Tchèque: Springer Berlin / Heidelberg, 2-6 Octobre 2006.

Suomi.fi. 2007. Taxonomy for Finnish services. http://demo.seco.tkk.fi/suomifi/?&m=0&g=&l=en

Tambouris, Efthimios, Stelios Gorilas, Gregory Kavadias, et coll. 2004. Ontology-enabled e-Gov service configuration : an overview of the OntoGov project. Dans *Knowledge Man-*

agement in Electronic Government (KMGov). Krems, Autriche: Springer, Berlin, 17-19 Mai 2004.

Tarby, Jean-Claude. 1993. Gestion automatique du dialogue homme-machine à partir de spécifications conceptuelles. Thèse de doctorat de l'Université Toulouse 1, France.

Tarby, Jean-Claude, et Marie-France Barthet. 1996. The DIANE+ method. Dans *CADUI*, 95-124. Namur, Belgique, 5-7 Juin 1996.

Tardieu, Hubert, Arnold Rochfeld, et René Colletti. 1983. *La méthode Merise - Tome 1 Principes et outils*. Paris, France: Editions d'organisation.

Tidwell, Jenifer. 2005. *Designing interfaces*. Sebastopol, California, USA: O'Reilly.

Tombelli, Richard. 2003. *Usability guidelines for e-Government applications*. Rapport d'études. Michigan, USA: Department of Information Technology, Juin 2003. http://www.michigan.gov/documents/Usability_guidelines_2003v1_72381_7.pdf

Troyer, Olga De, Sven Casteleyn, et Peter Plessers. 2008. WSDM: Web Semantics Design Method. Dans *Web Engineering: Modelling and Implementing Web Applications*, 303-351. Stanford, CA, USA: ACM, 11-12 Février 2008.

UK e-Government unit. 2004. *Guidelines for UK government websites: illustrated handbook for web management teams*.

Van der Aalst, W. M. P., et A. H. M. Ter Hofstede. 2002. *YAWL: Yet Another Workflow Language*. Rapport technique. Brisbane, Australie: Queensland University of Technology.

Van der Aalst, Wil, Arthur ter Hofstede, Bartek Kiepuszewski, et Ana Beatriz Barros. 2003. Workflow patterns. *Distributed and Parallel Databases* 14: 5-51.

Van der Veer, Gerrit C., Martijn Van Welie, et Cristina Chisalita. 2002. Introduction to Groupware Task Analysis. Dans *TAMODIA*, 32-39. Bucharest, Roumanie: INFOREC Publishing House, 18-19 Juillet 2002.

Van Duyne, Douglas K., James A. Landay, et Jason I. Hong. 2002. *The design of sites*. Addison-Wesley Professional, 22 Juillet 2002. http://www.designofsites.com/toc/index.htm (some of the chapters are available).

Van Welie, Martijn. 2002. Patterns for designers? Dans *Patterns Workshop (at CHI)*. Mineapolis, Minnesota, USA, 20-25 Avril 2002.

Van Welie, Martijn, et Gerrit C. Van der Veer. 2003. Pattern languages in interaction design: structure and organization. Dans *Interact*. Zürich, Suisse: IOS Press, 1-5 Septembre 2003.

Van Welie, Martjin. 2007. Web design patterns. *http://www.welie.com/patterns/*

Vanderdonckt, Jean, *Guide ergonomique des interfaces homme-machine*, Presses Universitaires de Namur, Namur, 1994, ISBN 2-87037-189-6.

Venkatesh, Viswanath, Michael G. Morris, Gordon B. Davis, et Fred D. Davis. 2003. User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly* 27, no. 3 (Septembre 2003): 425-478.

Verginadis, Giannis, Dimitris Gouscos, et Gregory Mentzas. 2004. Modelling e-Government Service Workflows through Recurring Patterns. Dans *EGOV*, 483-488. Zaragoza, Espagne: Springer Berlin, 30 Août – 3 Septembre 2004.

Victoria Online. 2006. *Web analytics toolkit*. Etude. Melbourne, Victoria, Australie, Septembre 2006. http://www.egov.vic.gov.au/pdfs/Web-Analytics-Toolkit-v1-Sept2006.pdf

Vuyst, Bruno de, et Alea Fairchild. 2006. The Phenix project: a case study of e-justice in Belgium. Dans *Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to*

conducting successful business on the internet, 327-333. Fredericton, New Brunswick, Canada: ACM, 13-16 Août 2006.

W3C. 2004a. *Resource Description Framework (RDF)*. Language Specification. W3C. http://www.w3.org/RDF/

---. 2004b. *OWL Web Ontology Language Overview*. Language Specification. W3C, Février 10. http://www.w3.org/TR/owl-features/

---. 2008a. Extensible Markup Language (XML). http://www.w3.org/XML/

---. 2008b. W3C XML Schema. http://www.w3.org/XML/Schema

---. 2008c. XSL Transformations (XSLT). http://www.w3.org/TR/xslt

---. 2008d. Cascading Style Sheets. http://www.w3.org/Style/CSS/

Wang, Lili, Stuart Bretschneider, et Jon Gant. 2005. Evaluating Web-Based E-Government Services with a Citizen-Centric Approach. Dans *HICSS*. Hawaii, USA: IEEE Computer Society, 3-6 Janvier 2005.

Watchfire. 2004. WebXACT. http://webxact.watchfire.com/

Wharton, Cathleen, John Rieman, Clayton Lewis, et Peter G. Polson. 1994. The Cognitive Walkthrough method: a practitioner's guide. Dans *Usability inspection methods*, 105-140. New York, NY, USA: John Wiley & Sons Inc.

White, Stephen. 2004. *Business Process Modeling Notation (BPMN)*. BPMI, Mars 2004.

Wild, Peter J., Peter Johnson, et Hilary Johnson. 2004. Towards a composite modelling approach for multitasking. Dans *3rd annual conference on Task models and diagrams*. Prague, République Tchèque: ACM, 15-16 Novembre 2004.

Wimmer, Maria A. 2001. Knowledge Management in e-Government - e-Commerce vs. e-Government. *http://falcon.ifs.uni-linz.ac.at/research/ceepus.zip*

---. 2006. Implementing a Knowledge Portal for e-Government Based on Semantic Modeling: The e-Government Intelligent Portal (eip.at). Dans *HICSS*. Kauai, Hawaii, USA: IEEE Computer Society, 4 Janvier 2006.

Winckler, Marco, Eric Barboni, Christelle Farenc, et Philippe Palanque. 2004. SWCEditor: a model-based tool for interactive modelling of web navigation. Dans *CADUI*, 55-66. Madeira Island, Portugal: Kluwer, 14-16 Janvier 2004.

Winckler, Marco, Christelle Farenc, Eric Barboni, et Florence Pontico. 2005. Modélisation orientée tâche de la navigation d'une application web : catalogue des thèses de l'AFIHM. Dans *IHM'05*. Toulouse, France: ACM, 27-30 Septembre 2005.

Winckler, Marco, et Philippe Palanque. 2003. StateWebCharts: a formal description technique dedicated to navigation modelling of Web applications. Dans *DSV-IS*. Madeira Island, Portugal: LNCS 2844 Springer, 11-13 Juin 2003.

Wood, Fred B., Elliot R. Siegel, Eve-Marie LaCroix, et coll. 2003. A Practical Approach to E-Government Web Evaluation. *IT Professional*, Mai 2003.

World Wide Web Consortium (W3C). 2007. Web Accessibility Initiative (WAI). http://www.w3.org/WAI/

Xiong, Joseph, Christelle Farenc, et Marco Winckler. 2007. Analyzing Tool Support for Inspecting Accessibility Guidelines During the Development Process of Web Sites. Dans *IWWUA workshop (at WISE)*, 470-480. Nancy, France: Springer Verlag, 3 Décembre 2007.

Yahoo! . 2007. Design pattern library. *http://developer.yahoo.com/ypatterns/*

Liste des figures

Liste des tables

Annexe A. Enchaînements d'écrans récurrents

ASSISTANT[1]

Lorsque plusieurs étapes sont nécessaires pour accomplir une procédure, un assistant est chargé de guider l'utilisateur au fil de ces étapes. L'assistant donne à chaque étape des indications sur le nombre d'étapes franchies (celles que l'utilisateur a validées et celles sur lesquelles il devra revenir), ainsi que les étapes restant à franchir. Dans l'exemple fourni sur la Figure 70, l'utilisateur de l'application Limosa[2] est en train de réaliser la troisième étape. Un récapitulatif est fourni en fin de séquence, afin que l'utilisateur puisse éventuellement revenir sur les informations qu'il a délivrées. Une fois le récapitulatif validé par l'utilisateur, un accusé de réception est diffusé, pour lecture simple ou impression à des fins de conservation pour preuve de la réalisation de la procédure.

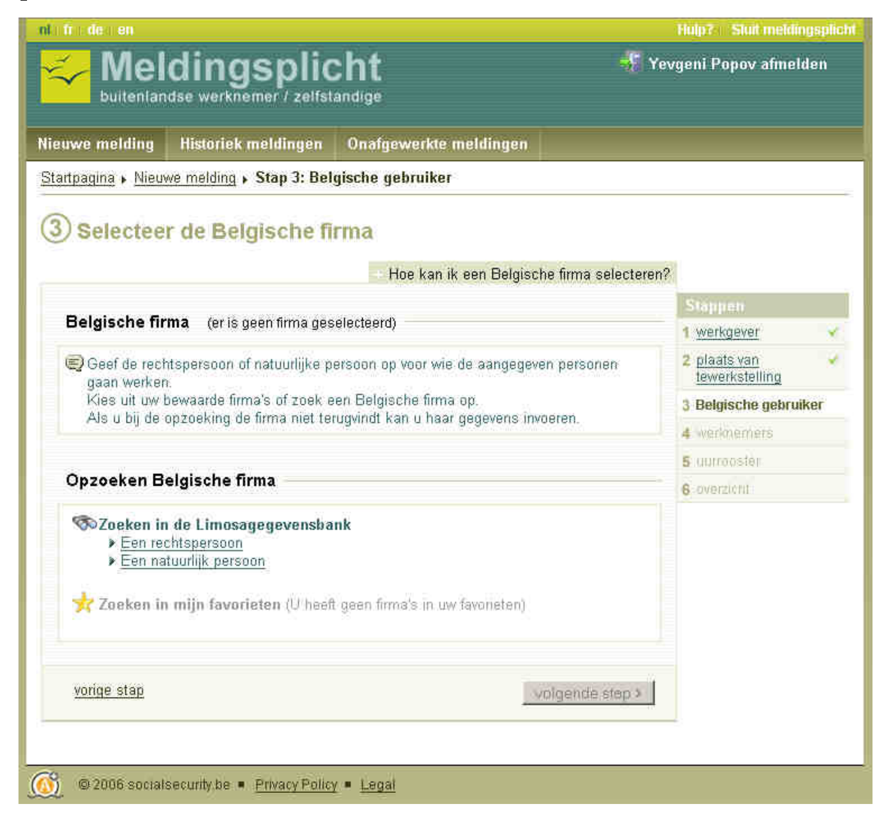


Figure 70. Étape d'un assistant sur le site Limosa (capture d'écran réalisée en octobre 2006)

Plusieurs enchaînements d'écrans coexistent pour cette famille d'applications. Dans un assistant avec guidage fort (cf. Figure 71), les étapes de la procédure doivent être réalisées dans l'ordre. Si l'utilisateur veut corriger les données saisies à l'étape i, il devra valider de nouveau les étapes i+1 jusqu'à la fin. Cette forme d'assistant est utilisée en particulier lorsque les données d'une étape sont dépendantes des données des étapes précédentes. Lorsque cette interdépendance entre données d'étapes successives est moins forte, mais que les concepteurs souhaitent tout de même fournir un guidage à l'utilisateur, qu'il conserve une représentation des données saisies comme étant fournies dans le cadre d'une procédure séquentielle, un assistant avec guidage souple peut être proposé (cf. Figure 72) : lors de l'édition du récapitulatif, un retour en arrière est proposé vers l'étape à modifier, suivi d'un retour direct au récapitulatif, sans repasser par les étapes suivant l'étape modifiée. Lorsque les données saisies sont

[1] En Anglais : *Wizard*

[2] http://www.limosa.be (dernier accès : 24 mars. 08)

totalement indépendantes d'une étape sur l'autre, l'assistant avec récapitulatif éditable est préférable, pour des modifications directes (cf. Figure 73). Ces deux derniers schémas de navigation, si possible, sont à privilégier pour les utilisateurs réguliers d'une application, notamment dans un cadre professionnel : la souplesse de l'application a alors plus d'importance que le guidage.

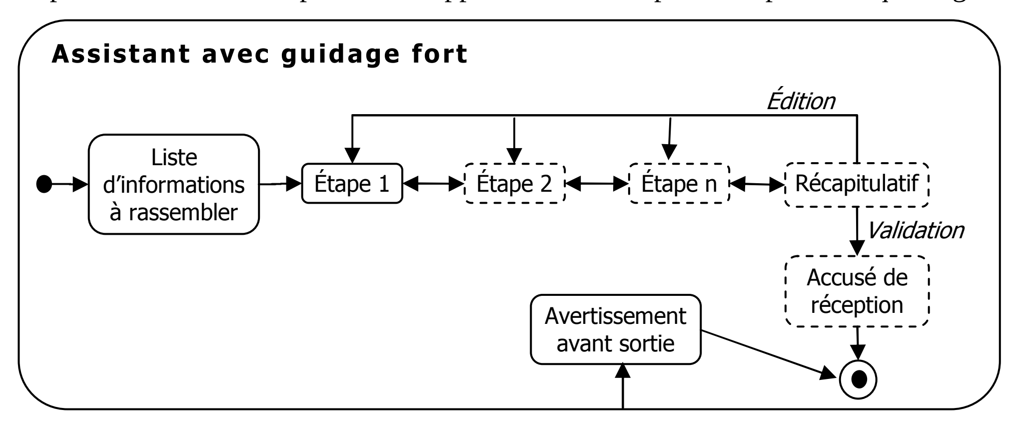


Figure 71. Modèle SWC de l'enchaînement d'écrans « Assistant avec guidage fort »

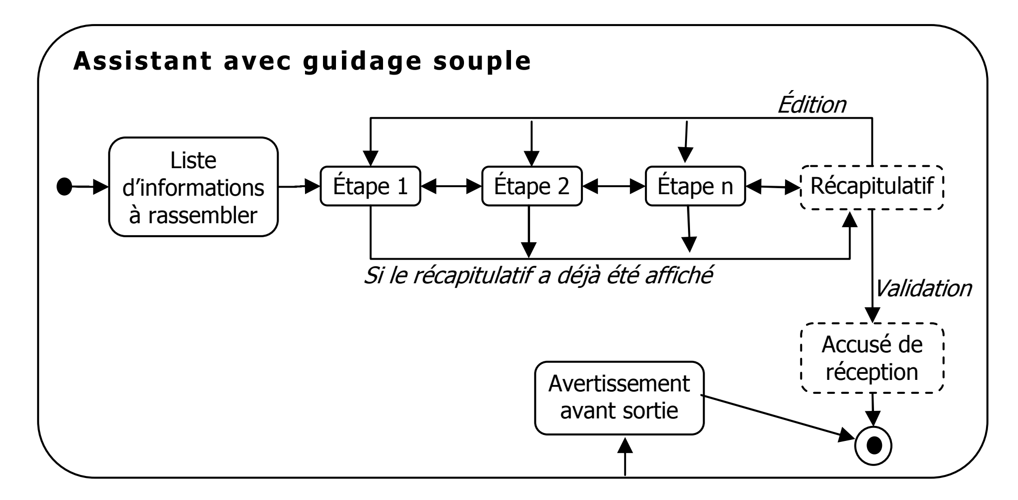


Figure 72. Modèle SWC de l'enchaînement d'écrans « Assistant avec guidage souple »

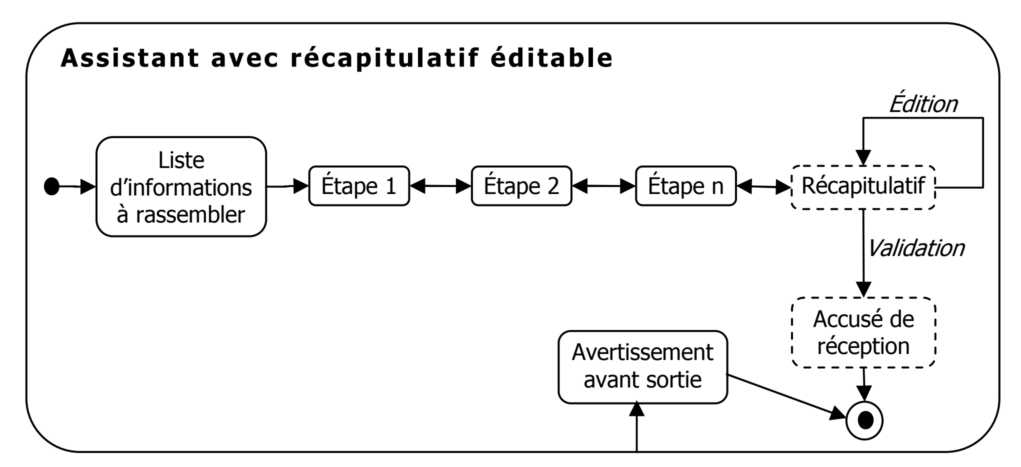


Figure 73. Modèle SWC de l'enchaînement d'écrans « Assistant avec récapitulatif éditable »

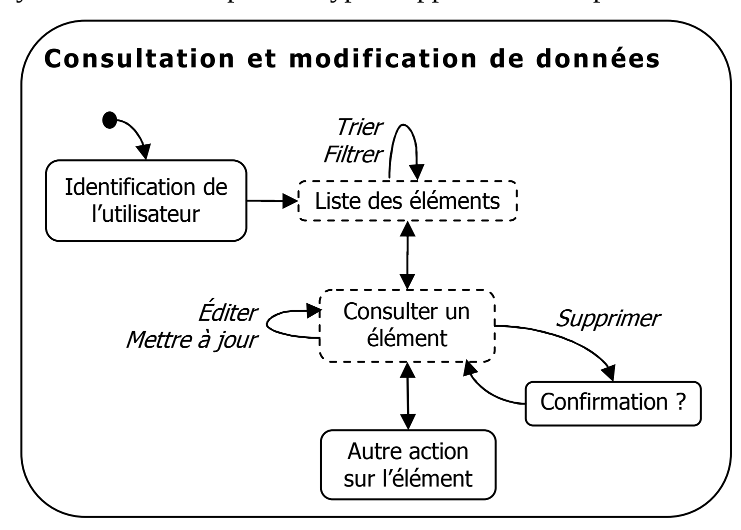
CONSULTATION ET MODIFICATION DE DONNÉES

Cet enchaînement d'écrans a été relevé sur de nombreuses applications, avec une complexité plus ou moins grande, selon le volume des données et leur nature. Un exemple est trouvé sur l'application PrimaWeb[1] qui permet aux employés du secteur social de consulter et de modifier des dossiers de suivi de personnes nécessitant une aide sociale (p.ex. ménages à faible revenu, réfugiés politiques). La capture d'écran de la Figure 74 concerne la consultation et l'éventuelle mise à jour de la composition d'un ménage déjà enregistré : mise à jour ou ajout d'un membre.



Figure 74. Complétion d'une enquête sociale via l'application PrimaWeb (capture d'écran extraite du manuel d'utilisation de l'application (BUCOM 2007))

Ce type d'enchaînement d'écrans concerne les applications où l'utilisateur est confronté aux données enregistrées dans une base au moment de l'utilisation. Des fonctions de tri ou de filtre sont généralement proposées, pour les bases de volume important. À partir de cette liste, l'utilisateur définit et/ou modifie les données enregistrées. Les données peuvent être de nature différente (p.ex. sur une application permettant la déclaration des revenus : coordonnées, revenus imposables, revenus non imposables, dons à des organismes caritatifs). D'autres opérations peuvent être mises à disposition de l'utilisateur en fonction de la nature des données et du rôle de l'utilisateur (p.ex. suppression de données, impression, établissement d'un document officiel à partir de ces données). Ces informations le concernent (p.ex. un citoyen modifie son adresse sur l'application Web de la sécurité sociale) ou concernent un tiers (p.ex. un chef d'entreprise déclare un nouvel employé à la sécurité sociale). L'enchaînement d'écrans généralement observé au fil de l'analyse de l'existante pour ce type d'applications est présenté sur la Figure 75.



[1] https://www.socialsecurity.be/site_fr/civilservant/Applics/primaweb/index.htm (dernier accès : 24 mars. 08)

Figure 75. Modèle SWC de l'enchaînement d'écrans « Consultation et modification de données »

GESTION DE DOSSIERS

Cet enchaînement d'écrans a été relevé sur les applications où les utilisateurs ont plusieurs dossiers à gérer en parallèle, sans rapport les uns avec les autres, mais de la même forme. L'utilisateur est généralement un professionnel dont l'utilisation de l'application est très régulière, voire quotidienne. Par exemple, les travailleurs sociaux ont plusieurs demandes d'aide à gérer en parallèle, chacune ayant son propre état d'avancement, via l'application PrimaWeb[1], comme présenté sur la Figure 76. Une nouvelle demande peut également être saisie, selon la même structure que les demandes déjà enregistrées. La sélection d'un dossier entraîne l'ouverture de sa description dans une nouvelle fenêtre, ce qui permet son traitement en parallèle du reste des activités de l'utilisateur.

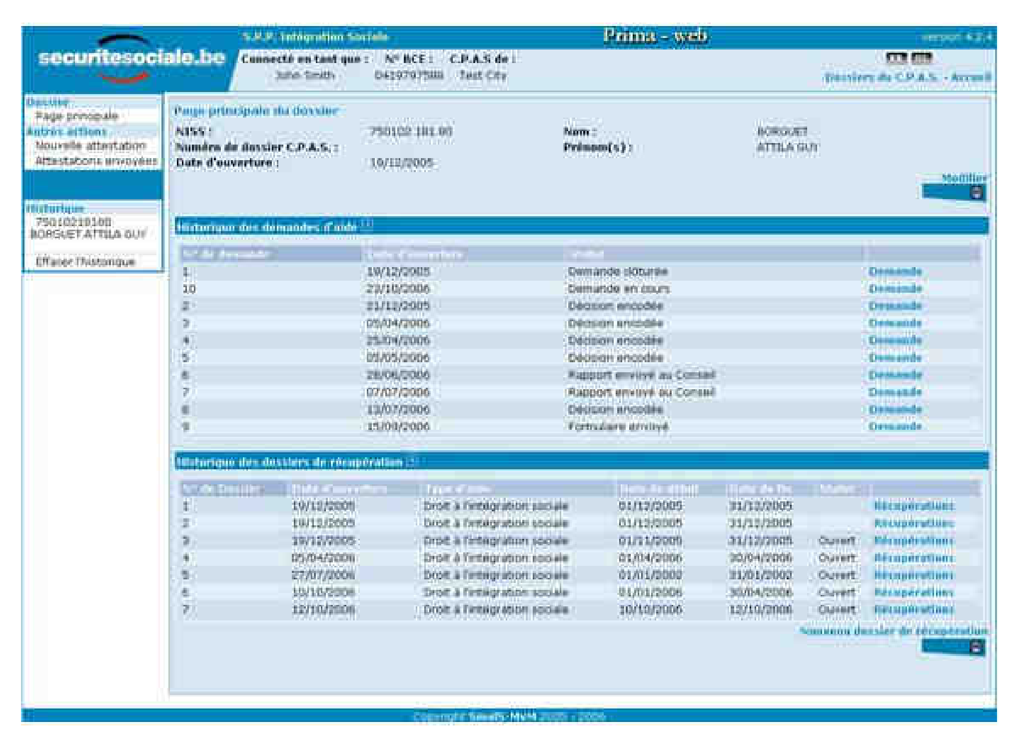


Figure 76. Liste des demandes d'aide à gérer via l'application PrimaWeb (capture d'écran extraite du manuel d'utilisation de l'application (BUCOM 2007))

Ce type d'applications requiert systématiquement l'identification de l'utilisateur, pour la sécurité des données à gérer, mais aussi pour connaître l'ensemble des données accessibles à l'utilisateur courant. Des fonctions de tri des dossiers sont généralement proposées, ainsi qu'une fonction d'ajout de dossier. Les dossiers sont gérés en parallèle les uns des autres, de sorte que les opérations peuvent être menées avec une grande liberté pour l'utilisateur. Pour permettre cela, l'interface est conçue de telle sorte qu'une nouvelle fenêtre s'ouvre pour chaque dossier à gérer, ou bien un système d'onglets est utilisé. Au moment de réaliser, sur plusieurs dossiers, une même tâche de mise à jour composée de trois opérations, un agent peut ainsi, s'il le souhaite, effectuer la première opération sur tous les dossiers, puis la seconde et enfin la troisième avant de valider. Il peut aussi réaliser la séquence d'opérations sur chaque dossier successivement, c'est à son libre choix. Cette latitude de comportement est particulièrement recherchée pour les applications à destination d'utilisateurs quotidiens d'une application, pour que soient respectés les modes de fonctionnement de chacun.

[1] https://www.socialsecurity.be/site_fr/civilservant/Applics/primaweb/index.htm (dernier accès : 24 mars. 08)

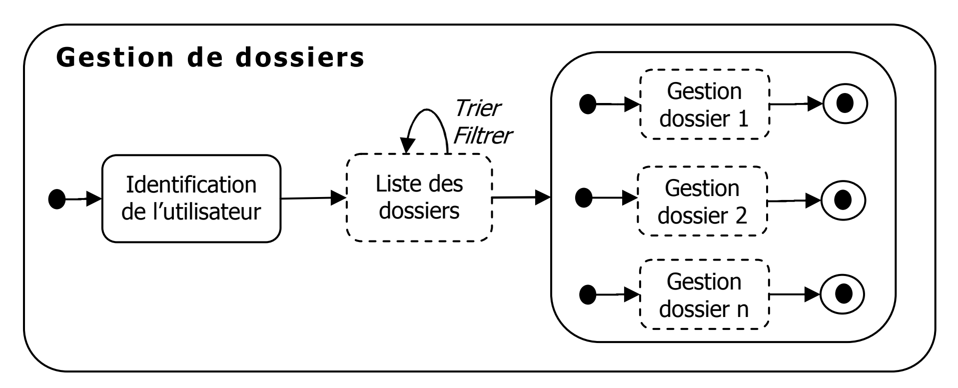


Figure 77. Modèle SWC de l'enchaînement d'écrans « Gestion de dossiers »

GESTION DE RÔLES

Lorsqu’une application diffuse des fonctionnalités pour plusieurs catégories d’utilisateurs, un principe de gestion des rôles doit être mis en place. L’utilisateur peut choisir le rôle qui lui convient par le biais de points d’entrée sur l’application, comme montré sur la Figure 78 : selon le cas, l’utilisateur choisit d’accéder à l’espace réservé aux citoyens, aux entreprises, aux professionnels de la sécurité sociale ou bien à l’espace international. Cet accès n’est pas soumis à identification, et rien n’empêche un citoyen de naviguer dans l’espace réservé aux professionnels : l’institution fait ainsi montre de transparence, et permet à un utilisateur curieux de connaître l’étendue de l’offre de l’application, au-delà des fonctionnalités qui le concernent directement. Si les pages décrivant les applications sont ainsi accessibles à tout visiteur, les applications proprement dites sont, en revanche, soumises à identification.

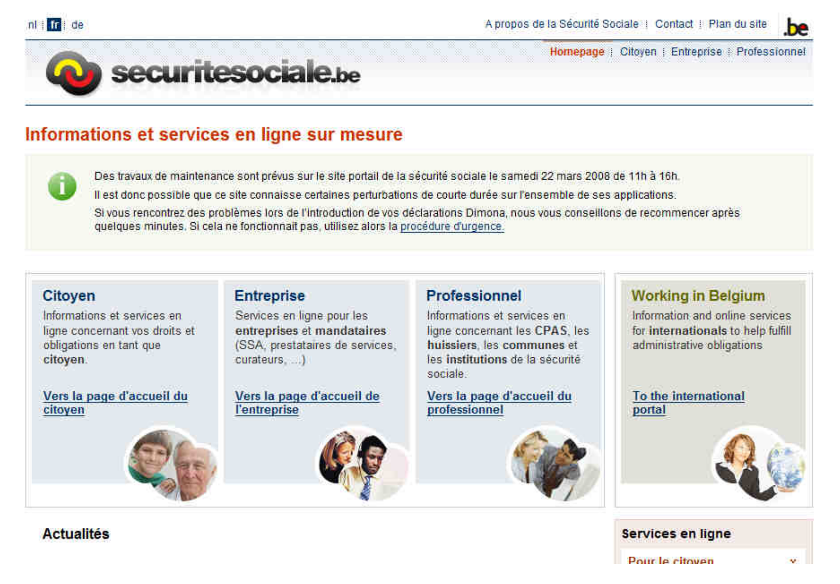


Figure 78. Page d'accueil du portail de la sécurité sociale belge (capture d'écran réalisée le 24 mars 2008)



Figure 79. En-tête des pages de l'espace consacré aux citoyens (capture d'écran réalisée le 24 mars 2008)

Ce mode de classification du contenu et des fonctionnalités d'une application d'e-Procuration donne à l'utilisateur novice l'impression d'obtenir une interface personnalisée, et de ne disposer que des fonctionnalités qui le concernent. L'épuration de la masse d'informations et de fonctionnalités permet ainsi à l'utilisateur d'avoir un accès plus direct à celles qui l'intéressent. Toutefois, d'autres parties de l'application peuvent le concerner, et un même utilisateur peut détenir plusieurs rôles (p.ex. un chef d'entreprise est aussi un citoyen) : le rôle courant doit apparaître clairement sur les pages visitées, ainsi que des accès directs à l'interface personnalisée pour les autres rôles (voir la Figure 79). Le modèle SWC de cet enchaînement d'écrans est présenté sur la Figure 80.

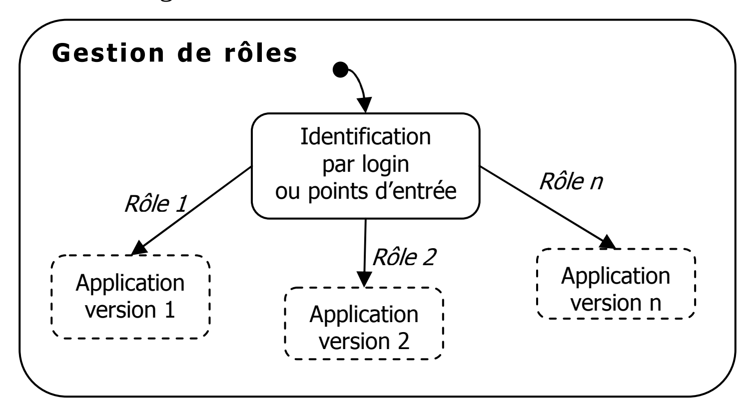


Figure 80. Modèle SWC de l'enchaînement d'écrans « Gestion de rôles »

ARCHITECTURE "HUB AND SPOKE"

Une architecture "*hub and spoke*" (littéralement « centre et rayon ») désigne traditionnellement un réseau de postes informatiques connectés entre eux via un noeud central. Un tel enchaînement d'écrans désigne un comportement de navigation où plusieurs fonctionnalités sont offertes à l'utilisateur comme autant de tunnels dans lesquels il entre, réalise des opérations, puis sort à la fin et retourne au point de départ. Contrairement à la séquence « Gestion de dossiers » étudiée ci-dessus, la réalisation d'une opération bloque les autres opérations ; l'utilisateur peut bien sûr réaliser plusieurs opérations, mais les unes après les autres, pas en parallèle.



Figure 81. Tableau de bord pour la gestion de la déclaration d'un travailleur à la sécurité sociale (capture d'écran issue du manuel de l'application DmfA (DmfA 2004))

L'application DmfA[1] de gestion des déclarations de travailleurs à la sécurité sociale est un exemple d'implémentation de ce comportement de navigation ; une capture d'écran du tableau de bord de cette application est présentée en Figure 81. Lorsque l'utilisateur choisit de modifier un des attributs du travailleur sélectionné, le formulaire de modification est ouvert à la place du tableau de bord, c'est-à-dire que toute autre activité avec l'application est impossible tant que celle-ci n'est pas terminée. Ce schéma de navigation permet d'offrir un guidage fort à l'utilisateur. Les opérations mises à disposition par le biais de tels schémas de navigation sont de courte durée, afin que l'utilisateur soit rapidement de retour sur le tableau de bord. Le modèle StateWebCharts de cette structure de navigation est présenté sur la Figure 82. Chaque opération peut être une page simple ou bien une séquence de pages.

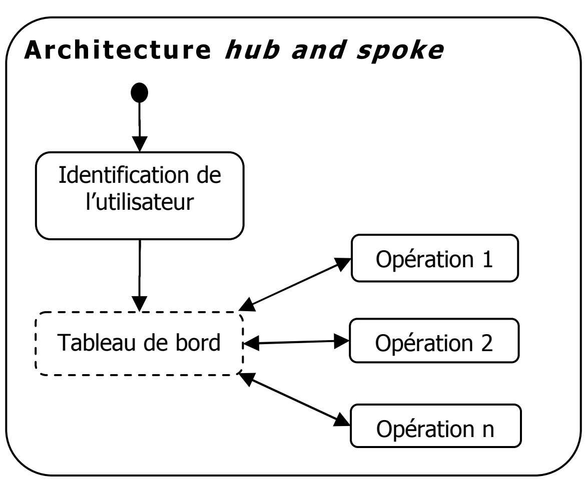


Figure 82. Modèle SWC de l'enchaînement d'écrans « Architecture *"hub and spoke"* »

INTÉGRATION À UN PORTAIL

L'intégration à un portail se fait généralement selon un schéma de navigation de type « Gestion de dossier », comme montré sur la Figure 83 pour l'application de gestion des interruptions de carrière et crédit temps[2]. Une nouvelle fenêtre est ouverte à chaque démarrage d'une application du portail. De cette façon, en dépit de l'apparence hétérogène des applications accessibles depuis le portail, la cohérence est gardée. La page du portail qui a permis de lancer l'application et qui contient les liens hypertextes essentiels au repérage (p.ex. lien vers la page d'accueil, lien vers plan du site) apparaît toujours en arrière plan.

[1] https://www.socialsecurity.be/site_fr/Applics/dmfa/index.htm (dernier accès : 25 sept. 08)
[2] https://www.socialsecurity.be/site_fr/civilservant/Applics/elo/index.htm (dernier accès : 24 mars. 08)



Figure 83. Ouverture d'une application à destination d'un professionnel dans une nouvelle fenêtre (capture d'écran réalisée le 24 mars 2008)

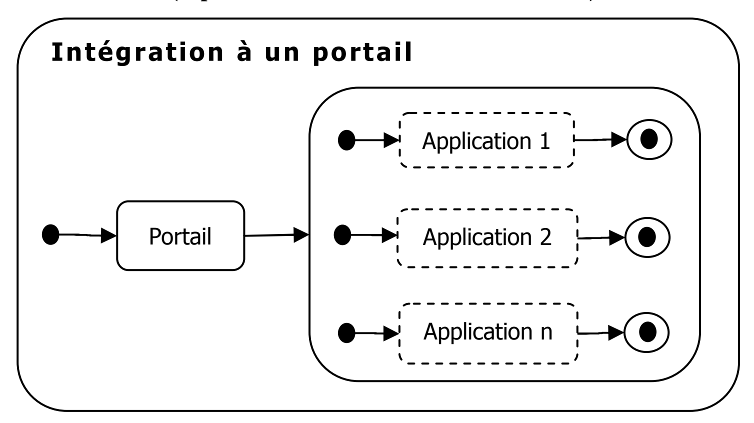


Figure 84. Modèle SWC de l'enchaînement d'écrans « Intégration à un portail »

Annexe B. Pages récurrentes

DIFFUSION DE DONNÉES

Les pages de cette section diffusent de l'information à l'utilisateur, dans un souci de clarté et de rigueur tout à la fois, certaines de ces pages constituant en effet une trace de la réalisation d'une procédure (p.ex. la page « Accusé de réception »). Cette information peut être statique, ou bien générée de façon dynamique, à partir d'une base de données liée à l'organisme concerné, ou à partir des informations préalablement collectées auprès de l'utilisateur.

- **Accusé de réception.** Quand une procédure a été réalisée avec succès, un accusé de réception est affiché à l'utilisateur, d'une part afin de le rassurer. L'utilisateur a été pris dans le tunnel d'une procédure en plusieurs étapes ; à l'issue de cette procédure, il est en effet important de lui préciser que cette procédure a abouti. D'autre part, dans le cas d'une procédure dont la réalisation est obligatoire (p.ex. déclaration d'un risque social à la sécurité sociale), des données d'horodatage figurent sur cet accusé de réception, pour que l'utilisateur puisse en conserver une trace concrète et parer à d'éventuels litiges. Un lien proposant l'impression de la page est souhaitable à ce niveau, comme il apparaît en haut à droite de la Figure 85.



Figure 85. Accusé de réception de l'application de déclaration de chômage temporaire (capture d'écran réalisée en octobre 2006)

- **Liste d'informations à rassembler.** Avant de commencer à réaliser une procédure, la liste d'informations à rassembler donne à l'utilisateur la possibilité d'anticiper ses actions et les documents dont il va avoir besoin (voir l'exemple de la Figure 86). Chacune des données nécessaire à la bonne marche de la procédure doit être clairement indiquée, accompagnée du moyen de la trouver (p.ex. figure sur un document, a été communiquée par email à une date précise). La durée de la procédure apparaît également, ainsi que des informations sur la pos-

sibilité de l’interrompre et de la reprendre ultérieurement. Les contraintes techniques éventuelles doivent également apparaître (p.ex. « Ce service en ligne requiert une version 1.5 ou plus récente de Mozilla Firefox ») ainsi que les moyens de les contourner (p.ex. « Téléchargez une version plus récente de votre navigateur http://www.firefox.fr1 »).

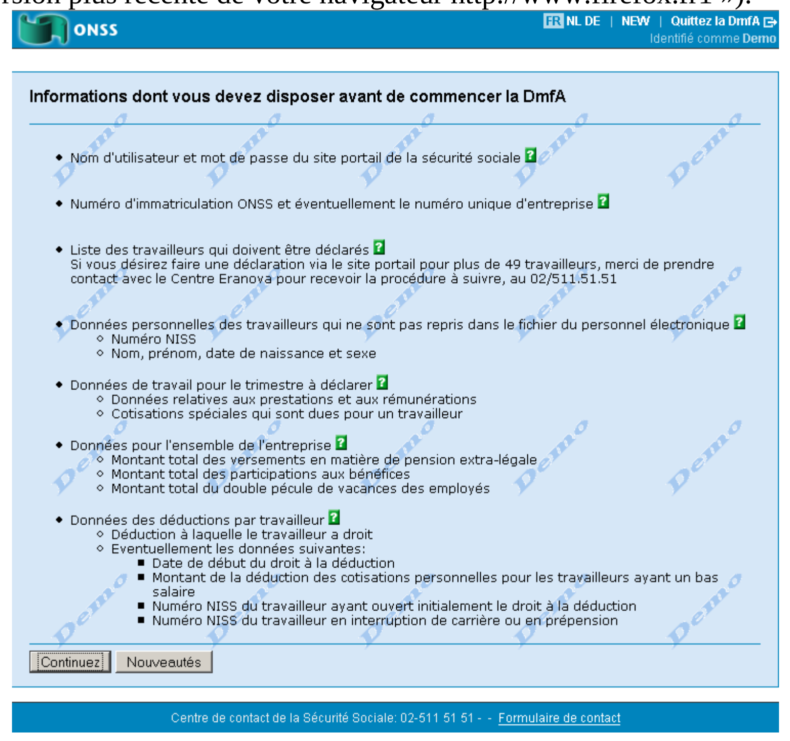


Figure 86. Liste d'informations à rassembler avant une déclaration via l'application de DmfA (capture d'écran réalisée en octobre 2006)

- **Données contextuelles.** Ces informations sont diffusées à l’utilisateur, généralement dans l’en-tête de la page, ou dans la colonne de gauche, pour apposer son empreinte et ainsi personnaliser l’interface. Son identité apparaît lorsqu’il s’est identifié personnellement, comme sur la Figure 87. Dans le cas de postes de travail partagés, cette marque de l’utilisateur identifié permet d’éviter l’utilisation accidentelle de l’espace de travail d’un autre utilisateur qui aurait oublié de se déconnecter. L’historique des opérations peut apparaître également, avec des liens vers des dossiers consultés antérieurement, ou bien des sections visitées. Ces données apparaissent en particulier dans le cas d’applications à destination de professionnels de l’administration, pour lesquels l’application est un outil de travail quotidien.

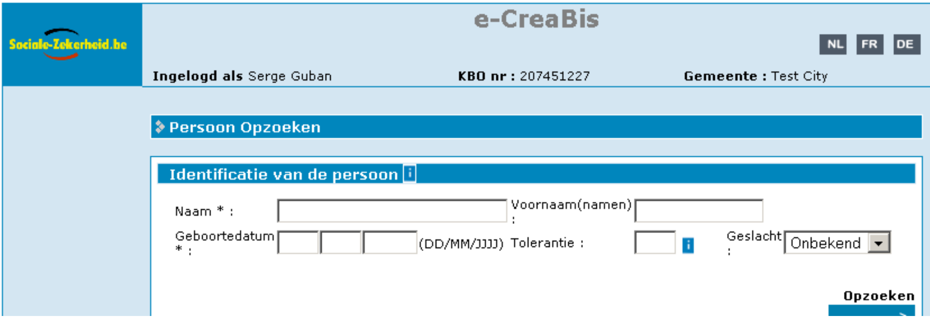


Figure 87. Données contextuelles disponibles sur l'application e-CreaBis (capture d'écran réalisée en octobre 2006)

[1] Dernier accès : 25 mars. 08

- **Récapitulatif.** Lorsque les étapes d'un assistant à la réalisation d'une procédure ont été réalisées avec succès, un récapitulatif est affiché. L'utilisateur est alors en mesure de confirmer ou de corriger certaines données saisies lors de la procédure. Pour le guider dans ces éventuelles corrections, il est conseillé de les repères de la procédure apparaissent encore, et les informations enregistrées au cours de la procédure apparaissent dans l'ordre de saisie. L'annulation totale de la réalisation de la procédure est encore possible, ou bien la correction d'une partie seulement des données saisies, directement sur le récapitulatif ou par le biais de retours en arrière vers les étapes à corriger, comme le propose le récapitulatif de la Figure 88.

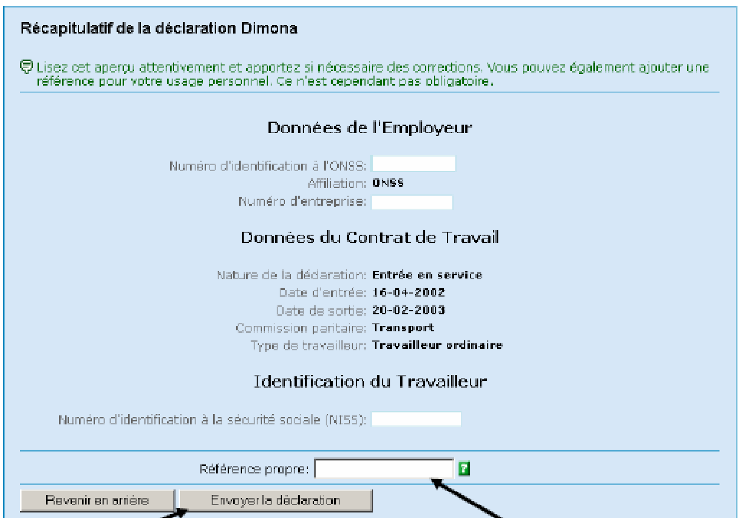
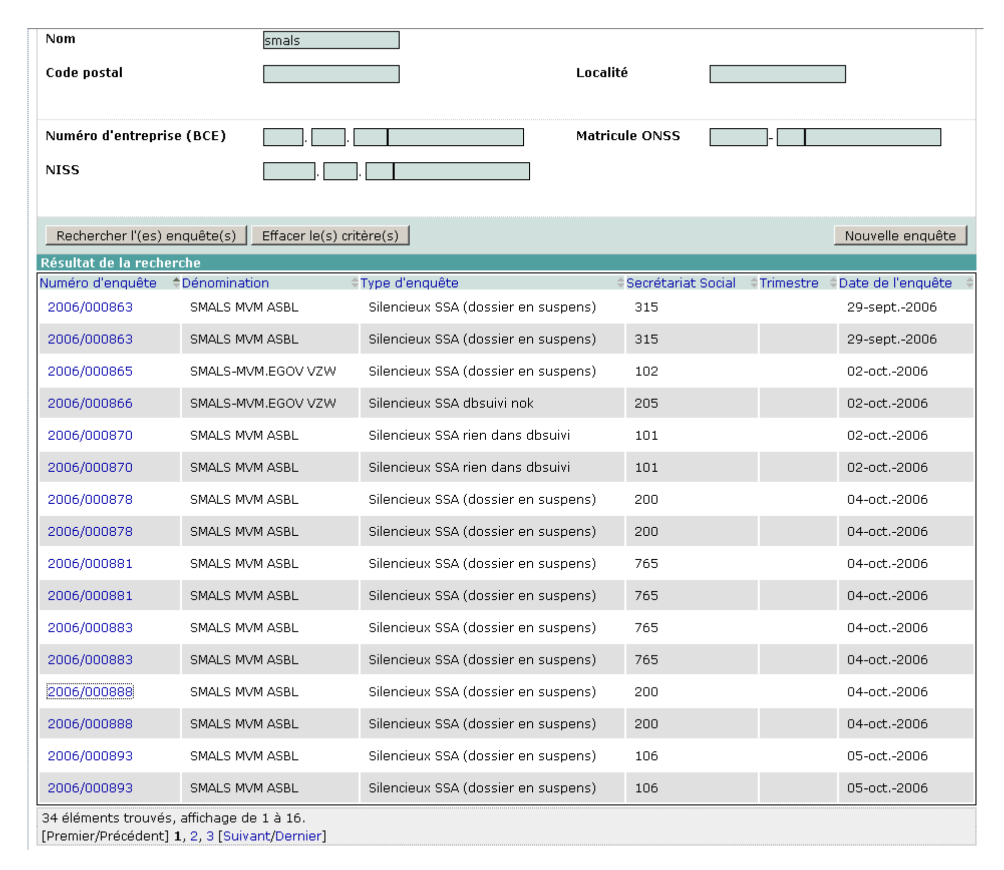


Figure 88. Récapitulatif de la déclaration Dimona
(capture d'écran réalisée en octobre 2006)

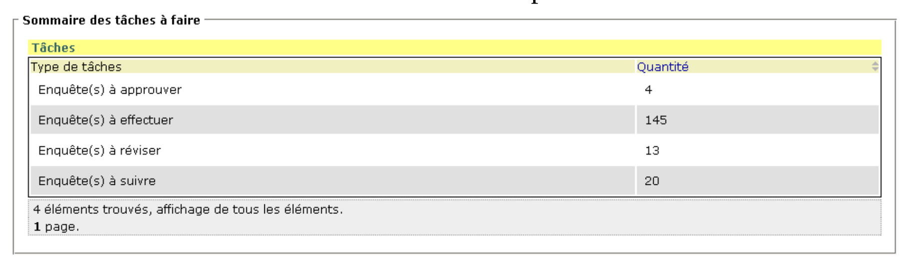
- **Liste d'éléments.** Une page fait souvent apparaître une liste d'éléments, pouvant atteindre de grands volumes dans le cas d'applications à destination de professionnels de l'administration. La liste présentée sur la Figure 89 est le résultat d'une recherche du nom « smals » dans les dossiers enregistrés sur l'application. Les éléments sont présentés dans un tableau, chacun sur une ligne, de couleur alternée pour faciliter la lecture. Les colonnes sont spécifiées par le responsable éditorial : il s'agit d'attributs porteurs de sens et fréquemment utilisés pour faire référence aux éléments listés.



Numéro d'enquête	Dénomination	Type d'enquête	Secrétariat Social	Trimestre	Date de l'enquête
2006/000863	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	315		29-sept.-2006
2006/000863	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	315		29-sept.-2006
2006/000865	SMALS-MVM.EGOV VZW	Silencieux SSA (dossier en suspens)	102		02-oct.-2006
2006/000866	SMALS-MVM.EGOV VZW	Silencieux SSA dbsuivi nok	205		02-oct.-2006
2006/000870	SMALS MVM ASBL	Silencieux SSA rien dans dbsuivi	101		02-oct.-2006
2006/000870	SMALS MVM ASBL	Silencieux SSA rien dans dbsuivi	101		02-oct.-2006
2006/000878	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	200		04-oct.-2006
2006/000878	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	200		04-oct.-2006
2006/000881	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	765		04-oct.-2006
2006/000881	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	765		04-oct.-2006
2006/000883	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	765		04-oct.-2006
2006/000883	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	765		04-oct.-2006
2006/000888	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	200		04-oct.-2006
2006/000888	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	200		04-oct.-2006
2006/000893	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	106		05-oct.-2006
2006/000893	SMALS MVM ASBL	Silencieux SSA (dossier en suspens)	106		05-oct.-2006

Figure 89. Liste d'éléments sur l'application iBoss (capture d'écran réalisée en octobre 2006)

♦ **Liste de tâches à réaliser.** Cette page aiguille l’utilisateur dans le choix de la tâche qu’il va accomplir avec l’application. Dans le cas d’applications à destination de professionnels de l’administration, comme dans la Figure 90, le volume de dossiers à la charge de l’utilisateur apparaît dès que l’utilisateur s’est identifié. Il a ainsi une vue d’ensemble sur la quantité de dossiers à traiter, et plus précisément sur les tâches à réaliser avec chacun, selon leur état d’avancement. Dans le cas d’une application à destination de citoyens (p.ex. déclaration de revenus), une telle page peut faire office d’aide mémoire, et agir en incitation à la réalisation de tâches dont la réalisation est limitée dans le temps.



Type de tâches	Quantité
Enquête(s) à approuver	4
Enquête(s) à effectuer	145
Enquête(s) à réviser	13
Enquête(s) à suivre	20

Figure 90. Liste de tâches à réaliser sur l'application iBoss (capture d'écran réalisée en octobre 2006)

♦ **Données d’horodatage.** Cette page contient les informations nécessaires à l’application pour l’identification de la tâche en cours, à venir ou passée. Par exemple, dans le cas de la Figure 91, la page est affichée en début de procédure. Elle est proposée à l’impression pour que l’utilisateur puisse donner une preuve de la réalisation de la procédure, ou obtenir une assistance efficace en cas de problème. Dans d’autres applications, ces données restent, comme données contextuelles, en tête de chaque page de la procédure, ou bien apparaissent seulement lorsque la procédure a abouti, sur l’accusé de réception.

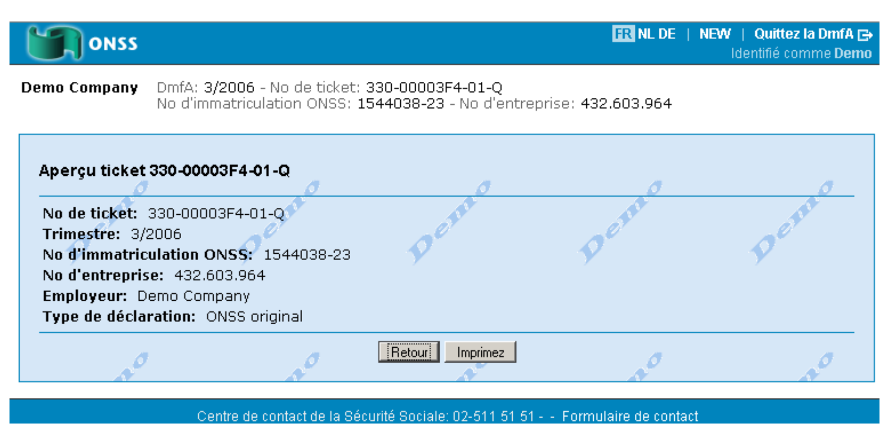


Figure 91. Données d'horodatage sur l'application DmfA (capture d'écran réalisée en octobre 2006)

OFFRE DE NAVIGATION

Sur le Web, l'utilisateur est face à un espace dans lequel le concepteur a lié sciemment des ressources les unes aux autres. Pour un espace de navigation efficace, l'utilisateur doit disposer de repères clairs sur sa localisation et les ressources auxquels il peut accéder. Dans le cas d'applications procédurales telles que celles analysées ici, les liens hypertextes peuvent valider ou invalider une activité, ou encore transmettre des données personnelles vers un serveur.

♦ **Points d'entrée directs.** Lorsque les tâches prévues par l'application sont peu nombreuses (jusqu'à sept), ou bien que quelques (jusqu'à sept) catégories d'utilisateurs sont attendus, la proposition de points d'entrée directs vers l'application est un mode de navigation efficace. Des points d'entrée peuvent également être proposés selon la langue à utiliser, cas de figure qui se produit quasi systématiquement en Belgique, pays dans lequel trois langues officielles coexistent. Dans le cas de la Figure 92, les points d'entrée concernent à la fois le statut de l'utilisateur (en recherche d'emploi ou responsable de recrutement dans une entreprise) et la tâche qu'il est susceptible de réaliser avec l'application (respectivement consulter une offre d'emploi ou en déposer une).



Figure 92. Points d'entrée directs sur l'application VDAB (capture d'écran réalisée en octobre 2006)

♦ **Page de tableau de bord.** Une fois identifié, l'utilisateur peut accéder à des informations personnelles qu'il peut modifier, ainsi qu'à de courtes procédures. La surcharge

d'informations est à éviter, en particulier dans le cas d'une application à destination de citoyens, pour conserver les avantages d'un tableau de bord qui aiguille directement sur la tâche ou la section recherchée, au premier coup d'œil. Les icônes et autres indicateurs graphiques sont utiles pour une lecture efficace de pareilles pages. Sur l'exemple fourni par la Figure 93, l'utilisateur est guidé dans le choix de sa tâche (ici le choix du dossier à compléter) par l'état d'avancement de la complétion des informations à saisir.

commencée et éventuellement encore adapter les données de travailleurs déjà remplies ou si vous avez choisi de commencer une déclaration sur base des données du trimestre précédent.

Nom	Prénom	NISS	Statut	Manières / Action
De Cock	Maarten	770319 249 43	Non complété	Simplifiée Détaillée
Eerdekens	Jan	790512 149 21	Non complété	Simplifiée Détaillée
Gistelinck	Katrien	750507 060 79	Non complété	Simplifiée Détaillée
Poppe	Thomas	771122 361 91	Non complété	Simplifiée Détaillée
Tilman	Michel	591005 049 43	Non complété	Simplifiée Détaillée

Modifier la sélection

Figure 93. Tableau de bord de l'application de DmfA (capture d'écran réalisée en octobre 2006)

- **Filtrage de liste.** Lorsque une liste d'éléments est diffusée à l'utilisateur, une fonctionnalité de filtrage permet d'alléger le volume d'éléments affiché, relativement au souhait de l'utilisateur. La sélection d'un ou plusieurs éléments est ainsi facilitée, et l'utilisateur évite les oublis dans cette sélection et dans les actions consécutives. La liste de travailleurs étrangers de la Figure 94 peut ainsi être filtrée suivant l'entreprise belge qui les emploie.



Figure 94. Liste d'éléments pouvant être filtrée, sur l'application Limosa (capture d'écran réalisée en octobre 2006)

- **Recherche.** Lorsque des éléments sont stockés dans une base de données, des attributs sont renseignés pour chacun d'eux, définis conjointement par les développeurs et les responsables éditoriaux. Ces attributs sont porteurs de sens pour l'utilisateur, leur valeur est un moyen de faire référence à un ou plusieurs objets de la base. Une recherche selon ces attributs permet à l'utilisateur d'obtenir un ensemble d'objets de caractéristiques spécifiques, comme par exemple, pour la Figure 95, un ensemble d'enquêtes concernant une entreprise donnée.

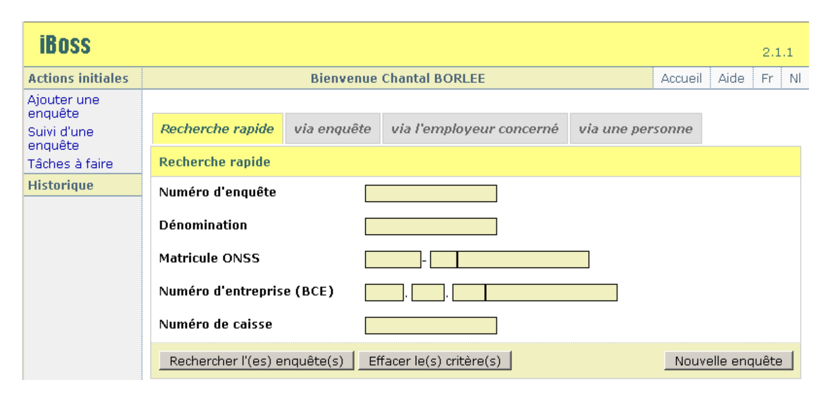


Figure 95. Recherche d'éléments sur l'application iBoss (capture d'écran réalisée en octobre 2006)

♦ **Navigation de service.** Les liens dont l'utilisateur aura probablement besoin au cours de sa visite d'une application Web peuvent apparaître en tête et au pied des pages de l'application (plus rarement dans la colonne de gauche), de façon constante pour un repérage facile. Dans le cas de l'application de la Figure 96, et comme souvent dans les applications analysées, apparaissent les liens qui sont souvent visitées (agenda, liens, recherche), des informations sur l'identité de l'application (logo, plan du site, contact), et des liens concernant le profil de l'utilisateur (identification, déconnexion).



Figure 96. Navigation de service sur l'application VDAB (capture d'écran réalisée en octobre 2006)

♦ **Tri de liste.** Les listes d'éléments proposées sont souvent associées à des fonctionnalités de tri sur les attributs des éléments qui apparaissent en tête des colonnes. Dans le cas de la Figure 97, les éléments stockés sur l'application courante sont des documents produits par d'autres applications d'e-Gouvernement. Ils peuvent être triés selon leur description, leur taille ou encore leur date de création, de façon croissante ou décroissante. Le mode de tri en cours doit être indiqué clairement ; ici il est indiqué seulement pour le mode de tri dans le-

quel les documents lus (respectivement non lus) apparaissent en tête, grâce à la flèche d'en-tête de la deuxième colonne.

Reçus | Corbeille

Rechercher dans «Tous», les documents correspondant aux critères suivants: Cacher l'écran de recherche

Critères généraux

Numéro de ticket: Lus: ☑

Reçus depuis: 12/4/2006 Non-lus: ☑ Demandé par:

RECHERCHER EFFACER

		Description	Date de création	Date d´expiration	Application	Demandé par	Taille
☐	✉ i	C4 - Certificat de chômage pour GUY RENE TONIEL...	25/10/2006	23/04/2007	DRS	SMALS-MVM	27
☐	✉ i	Déclaration simplifiée d´un accident du travail...	19/10/2006	17/04/2007	DRS	SMALS-MVM	10
☐	✉ i	Feuille de renseignements - Incapacité de trava...	18/10/2006	16/04/2007	DRS	SMALS-MVM	24
☐	✉ i	Salaires et temps de travail après l´accident d...	17/10/2006	15/04/2007	DRS	SMALS-MVM	11
☐	✉ i	Salaires et temps de travail après l´accident d...	17/10/2006	15/04/2007	DRS	SMALS-MVM	11
☐	✉ i	Vakantieattest voor bedienden voor ELS VERVOORT...	12/10/2006	10/04/2007	DRS	SMALS-MVM	20
☐	✉ i	340-000ZMWB-00-B (20)	12/10/2006	10/04/2007	DRS	SMALS-MVM	7
☐	✉ i	C3.2 (maandelijks) - economische redenen voor M...	11/10/2006	09/04/2007	DRS	SMALS-MVM	22
☐	✉ i	C4 - Werkloosheidsbewijs voor MARIE PAULE DELPO...	11/10/2006	09/04/2007	DRS	SMALS-MVM	27
☐	✉ i	Vereenvoudigde aangifte van een arbeidsongeval ...	11/10/2006	09/04/2007	DRS	SMALS-MVM	10

VERS CORBEILLE

<< 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 >>

QUITTER AIDE

Figure 97. Liste d'éléments pouvant être triée, sur l'application *e*-Box (capture d'écran réalisée en octobre 2006)

ENTRÉES DE L'UTILISATEUR

Les applications d'e-Procuration requièrent la participation de l'utilisateur en termes de saisie de données. Qu'il s'agisse d'un formulaire simple ou en plusieurs étapes, et que ce formulaire soit destiné à recueillir les remarques de l'utilisateur sur sa visite sur l'application, ou qu'il soit destiné à réaliser une procédure administrative, sa présentation doit être conforme aux aptitudes de l'utilisateur. L'enjeu de l'accessibilité de l'application par le biais de tout équipement et à toute personne se joue ici en grande partie.

- **Formulaire de contact.** L'utilisateur, via cette page, peut envoyer un message à l'institution concernée par l'application, pour poser une question ou déposer une remarque concernant l'application elle même. Dans le cas des applications du portail de la sécurité sociale, comme le montre la Figure 98, c'est le centre d'assistance à l'utilisateur interne à SmalS qui est contacté pour répondre aux attentes des utilisateurs. Ces contacts sont tracés pour d'éventuelles mises à jour des applications pour lesquelles des remarques récurrentes seraient relevées.

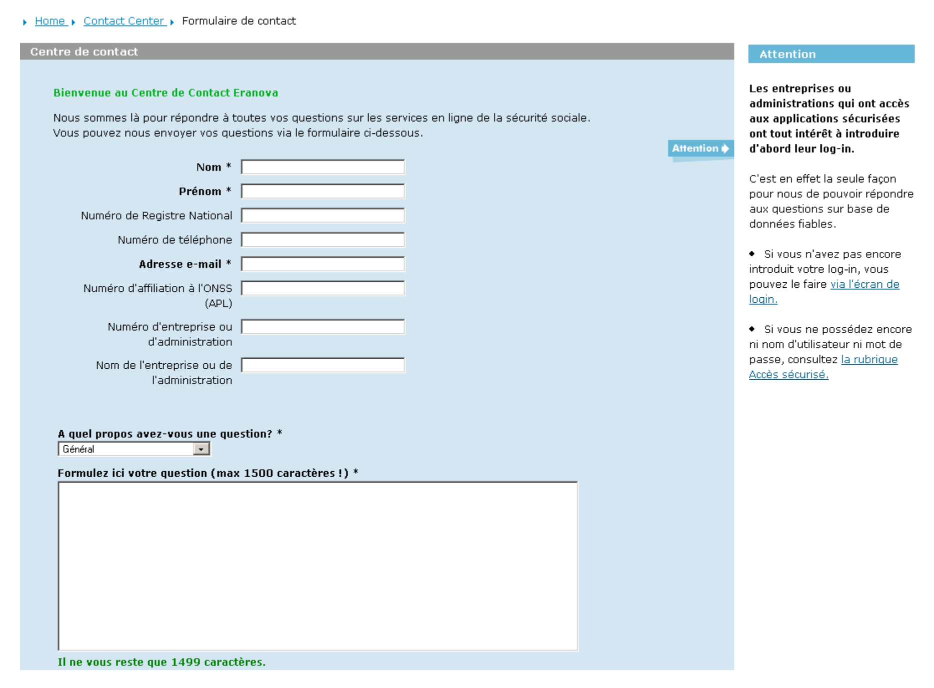


Figure 98. Formulaire de contact du portail de la sécurité sociale (capture d'écran réalisée en octobre 2006)

- **Formulaire de déclaration.** Les formulaires simples, dits de déclaration, contiennent un ensemble de champs qui, par souci de lisibilité, sont groupés par catégorie dès que leur nombre dépasse sept. Les en têtes de groupement de champs doivent être clairement signalés, comme par exemple dans l'exemple de la Figure 99 où ils apparaissent sur fond bleu. Les contraintes de saisie doivent apparaître de façon lisible et incitative : si un champ est défini comme requis par le concepteur du formulaire, la mention « zone obligatoire » apparaît par le biais d'un astérisque à la suite du champ concerné ; lorsqu'un format particulier est attendu, il doit apparaître à la suite du champ (p.ex. format de date JJ/MM/AAAA ou JJ/MM/AA).

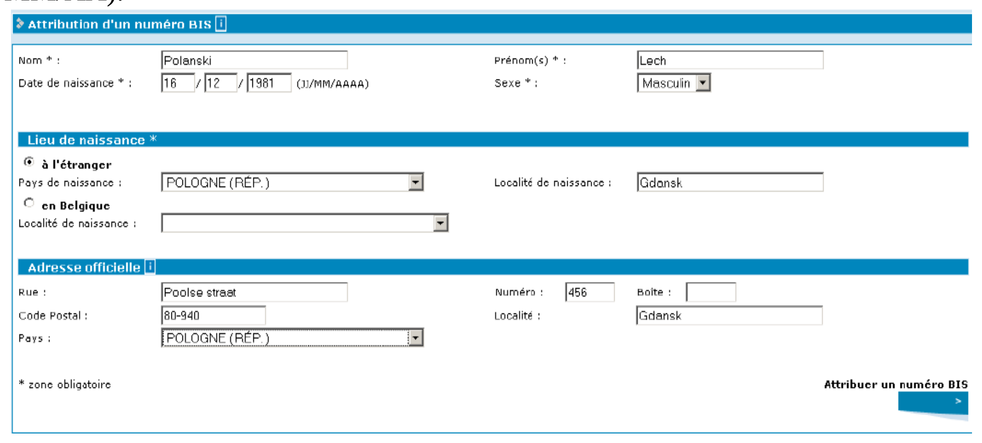


Figure 99. Formulaire de déclaration de l'application *e*-CreaBis (capture d'écran réalisée en octobre 2006)

- **Page d'identification.** Elle permet à l'utilisateur de s'identifier, pour continuer une procédure mise en pause, pour compléter les données déjà saisies, pour accéder à son espace personnel et, de façon générale, pour accéder aux fonctionnalités associées à son rôle. Comme indiqué sur la Figure 100, plusieurs modes d'identification sont offerts à l'utilisateur : une

identification par *token*[1] délivré par l'administration ou directement via un lecteur de carte d'identité électronique. Dans la mesure où elle ne nécessite pas l'envoi d'un identifiant par l'administration au préalable, la carte d'identité électronique est privilégiée pour les démarches à titre civil, de façon volontaire et ponctuelle. Elle contient un certificat permettant l'authentification de l'utilisateur et un autre pour la signature électronique.

Vous avez le choix de vous identifier d'une des deux manières suivantes:

- soit en utilisant votre user-ID, mot de passe et token citoyen, reçu via votre enregistrement sur le site www.belgium.be

  Identification par token.

- soit en utilisant votre carte d'identité électronique, en cliquant sur le lien ci-dessous.

  Identification par carte d'identité électronique.

Plus d'informations sur la carte d'identité électronique.

Figure 100. Page d'identification du portail de la sécurité sociale (capture d'écran réalisée en octobre 2006)

- **Étape d'assistant.** Lorsque plusieurs étapes sont à effectuer pour réaliser une procédure, un assistant guide l'utilisateur à travers ces étapes. Chaque étape présente une forme similaire, pour aider l'utilisateur à se repérer. Le titre de la procédure apparaît tout au long des étapes, et est construite de préférence autour d'un verbe d'action correspondant au but de la procédure. En bas de page, les boutons « Précédent » et « Suivant » lui permettent de naviguer pas à pas dans ces étapes. Une boîte d'avancement apparaît sur le côté droit de la page, comme par exemple sur la Figure 101. Les étapes franchies, avec succès ou insuccès, sont indiquées clairement, et l'utilisateur se localise ainsi au cœur de la procédure.



Figure 101. Étape d'assistant de l'application Limosa (capture d'écran réalisée en octobre 2006)

GUIDAGE

Les opérations de guidage se limitent ici au remplissage des formulaires, et laissent de côté les pages générales d'aide à l'utilisateur. Il est toutefois à noter que tout élément d'interface, qu'il s'agisse d'un élément d'affichage, de navigation ou de saisie, comporte sa part de guidage.

[1] Jeton

Toutefois, dans le cas d'un formulaire, l'affichage des contraintes pour la saisie d'un formulaire ne met pas à l'abri d'une erreur de l'utilisateur, qu'il s'agisse de l'oubli d'un champ ou d'une erreur syntaxique. Dans le cas de certaines applications, des erreurs sémantiques peuvent être définies également (p.ex. le congé maternité ne peut pas concerner un homme). Les concepteurs des pages analysées au fil de notre étude ont pris soin de mettre en avant les champs concernés par les erreurs : référence au label du champ dans le message d'erreur, mise en surbrillance du champ concerné, comme dans l'exemple de la Figure 102. Les erreurs peuvent être indiquées, comme dans ce même exemple, en tête de page, ou bien par une fenêtre contextuelle[1] modale.

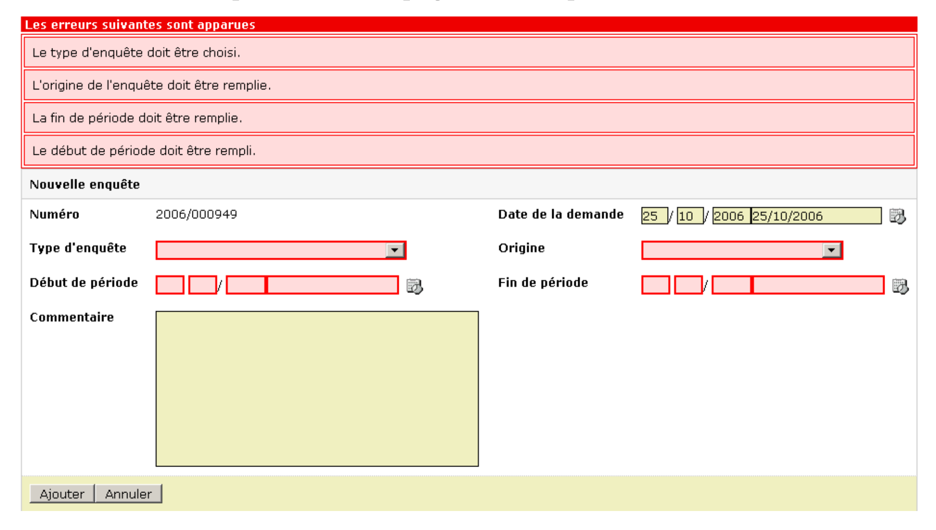
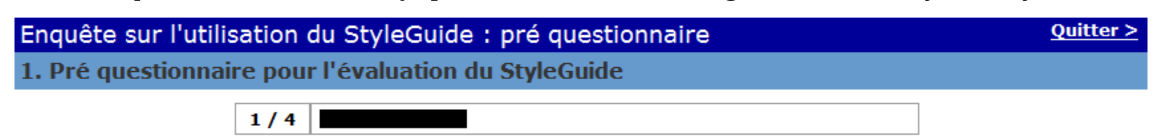


Figure 102. Alerte sur des erreurs de saisie d'un formulaire de l'application iBoss (capture d'écran réalisée en octobre 2006)

[1] En Anglais : *pop-up*

Annexe C. Questionnaire d'évaluation du StyleGuide

Ce questionnaire a été conçu par le biais de la version gratuite de SurveyMonkey[1].


Enquête sur l'utilisation du StyleGuide : pré questionnaire Quitter >

1. Pré questionnaire pour l'évaluation du StyleGuide

1 / 4

Ce questionnaire ouvre une enquête sur l'utilisation du StyleGuide au sein de SmalS. L'objectif est d'améliorer les conditions d'utilisation du StyleGuide. Les données sont collectées dans le but de cerner le profil des utilisateurs du StyleGuide et de comprendre leurs modes d'usage. Les données brutes récoltées ne seront en aucun cas divulguées à un tiers.

L'enquête se déroulera en deux étapes :
1) Remplissage du présent questionnaire (durée : moins de 15 minutes)
2) Entretiens complémentaires, du 15 au 18 avril 2008, avec certaines des personnes ayant rempli le questionnaire (durée : environ 1 heure)

Les résultats de cette enquête seront exploités dans le cadre de ma thèse de doctorat, qui porte sur le développement d'une méthode de conception pour l'e-Gouvernement. Mes travaux ont été étayés par un stage d'un mois chez SmalS en Octobre 2006 : j'avais alors eu l'opportunité de participer aux réflexions de conception du StyleGuide.

Cette enquête sera d'une très grande utilité pour appuyer ma réflexion et valider ma contribution. Je vous remercie sincèrement de l'aide que vous m'apportez en acceptant d'y participer.

Bien cordialement,
Florence Pontico, IHCS (Toulouse, France)
http://www.irit.fr/~Florence.Pontico

Suivant >

[1] http://www.surveymonkey.com (dernier accès : 18 avr. 08)

Enquête sur l'utilisation du StyleGuide : pré questionnaire

Quitter >

2. Mon profil

2 / 4

1. Je suis

- un homme
- une femme

J'indique mon âge ci-dessous :

2. J'utilise les outils informatiques aux fréquences suivantes

	jamais	plus rarement	plusieurs fois par mois	plusieurs fois par semaine	tous les jours
Editeur de schémas (ex. Microsoft Office Visio)					
Editeur de pages HTML et CSS (ex. FrontPage, Dreamweaver)					
Navigateur Internet (ex. Mozilla Firefox, Internet Explorer, Netscape)					
Bureautique (ex. Microsoft Office Word, PowerPoint, Excel, Suite OpenOffice)					

3. Mon statut principal est

- Chef de projet
- Analyste
- Consultant en utilisabilité
- Membre du Usability Center
- Graphiste
- Développeur d'interface
- Développeur
- Autre statut

4. Je travaille au contact de SmalS depuis

- moins de 6 mois
- de 6 mois à 1 an
- de 1 an à 3 ans
- plus de 3 ans

5. Ma connaissance des langues

	Je ne connais pas cette langue	J'ai des notions élémentaires	Je peux tenir une conversation	Je maîtrise cette langue
Français				
Néerlandais				
Anglais				

< Précédent | Suivant >

Enquête sur l'utilisation du StyleGuide : pré questionnaire — Quitter >

3. Mon expérience du StyleGuide

3 / 4

6. Mon initiation au StyleGuide

	oui	non
J'ai participé à une formation au StyleGuide		
Je sais à qui demander conseil pour d'éventuels problèmes d'usage du StyleGuide		
J'ai les compétences nécessaires pour utiliser le StyleGuide		
Le StyleGuide s'intègre bien à mes habitudes de travail		

Un commentaire à propos de ma prise en main du StyleGuide ? Circonstances particulières ? Difficultés rencontrées ?

7. Les activités que j'ai menées avec le StyleGuide

	jamais	1 fois	2 ou 3 fois	4 fois et plus
J'ai participé à un projet utilisant le StyleGuide				
J'ai exploré le StyleGuide sans but, par simple curiosité				
J'ai exploré le StyleGuide à la recherche d'un problème précis				
J'ai laissé un commentaire à propos d'un patron				
J'ai personnalisé un wireframe du StyleGuide				
J'ai argumenté une décision de conception en faisant référence au StyleGuide				

8. Qui, autour de moi, utilise le StyleGuide ?

- [ ] Mes responsables
- [ ] Mes subalternes
- [ ] Mes collègues de bureau
- [ ] Le client (il l'utilise ou est au courant de son existence)
- [ ] Aucun des choix précédents

9. Mon avis subjectif sur le StyleGuide

	pas du tout d'accord	plutôt pas d'accord	plutôt d'accord	tout à fait d'accord
J'utiliserais le StyleGuide même si j'avais une autre alternative				
Je conseillerais à mes collègues d'utiliser le StyleGuide				
Utiliser le StyleGuide améliore la qualité de mon travail				
Utiliser le StyleGuide accélère mes activités de travail				
Ceux qui utilisent le StyleGuide gagnent en considération				
Les problèmes de conception d'interface sont décrits avec clarté				
Les conseils de conception sont utiles				
Trouver un patron dans le StyleGuide est facile				
Comprendre un patron est facile				
Appliquer un patron est facile				
Quand plusieurs patrons pourraient répondre au problème de conception, les comparer est facile				

Une autre impression générale sur le StyleGuide ?

< Précédent | Suivant >

Enquête sur l'utilisation du StyleGuide : pré questionnaire — Quitter >

4. Mes coordonnées

4 / 4

10. Merci d'avoir pris le temps de répondre à ce questionnaire !

Pour finir, indiquez une adresse email à laquelle nous pourrons vous joindre pour un éventuel entretien d'approfondissement.

< Précédent | Valider >

Annexe D. Matériel d'évaluation du StyleGuide

ACCORD DE CONSENTEMENT ÉCLAIRÉ

Par la présente, le participant identifié au point 2. accepte de laisser enregistrer sa discussion avec l'évaluateur ainsi que ses interactions avec l'ordinateur mis à sa disposition dans le cadre de l'expérience décrite au point 1. Les parties 3. et 4. s'engagent à la confidentialité des données enregistrées et à ne pas utiliser ces données à d'autres fins que celles en rapport avec la recherche décrite au point 1.

1. Description de l'expérience

Evaluation des conditions d'usage du StyleGuide et de l'utilisabilité du formalisme StateWebCharts dans le cadre de la conception de l'interface d'une application d'e-Gouvernement.

2. Le Participant

Nom ……………………………………

Prénom ………………………………

Date de participation ………………

N° de participation …………………

Adresse ………………………………

……………………………………………

……………………………………………

Tél. ……………………………………

Mail …………………………………

Signature

3. Le Responsable de l'expérience

Nom Kanellopoulos

Prénom Garifalia

Société SmalS

Adresse Rue du Prince Royal 102

1050 Bruxelles

Belgique

Signature

4. L'observateur

Nom Pontico

Prénom Florence

Société IHCS – IRIT

(Université Paul Sabatier)

Adresse 118 Route de Narbonne

31062 Toulouse cedex 9

France

Signature

ENTRETIEN PRÉLIMINAIRE

Mon statut
☐ Analyste
☐ Développeur
☐ Editeur de contenu
☐ Graphiste
☐ Autre, je précise : ______________________________

J'ai rempli le pré questionnaire sur le Web
☐ oui
☐ non : je le remplis maintenant

Pour consulter le StyleGuide
☐ J'y accède sur le Web
    ☐ Il est dans mes favoris
    ☐ Autre moyen d'y accéder, je précise : ______________________________

☐ J'ai imprimé les guidelines les plus utiles
    Je précise lesquels, de mémoire : ______________________________
    ____
    ____

☐ Autre, je précise : ______________________________

Qui a présenté la 1ère page HTML dans les projets avec StyleGuide ?
☐ Un analyste
☐ Un graphiste
☐ Un développeur
☐ Autre, je précise : ______________________________

J'ai noté des différences entre les projets avec/sans StyleGuide
☐ Oui : avec le StyleGuide,
    ☐ Les pages HTML ont été présentées plus tard
    ☐ Des arguments ont fait référence au StyleGuide
    ☐ Le StyleGuide a été projeté lors de réunions
☐ Non

A mon avis, le principal avantage du StyleGuide est qu'il permet de :

SCÉNARIO 1

L'application Web suivante en est au stade d'analyse fonctionnelle.

- **Site web de la cantine de l'école communale de Goville**
- Accès par identifiant pour un parent pour inscrire ses enfants à la cantine certains jours (l'identifiant est remis en mains propres)
- Un repas peut être ajouté ou annulé jusque trois jours avant la date du repas
- Les responsables de la cantine peuvent consulter le nombre de repas prévus le jour même ou les jours à venir

Proposer, avec l'aide du StyleGuide, un prototype de la page permettant à un parent de consulter et modifier les dates de repas d'un de ses enfants.

SCÉNARIO 1BIS

StateWebCharts est une notation formelle de type « diagramme d'états – transitions » permettant de décrire la **structure hypertexte** d'une application Web.
Nous n'utilisons ici qu'une partie de la notation, celle qui considère le **point de vue de l'utilisateur** (en laissant de côté les actions côté serveur).

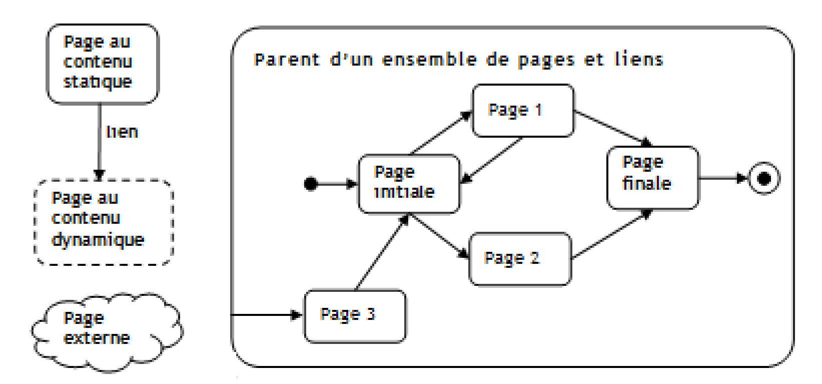


Figure 1. Éléments de base du formalisme StateWebCharts

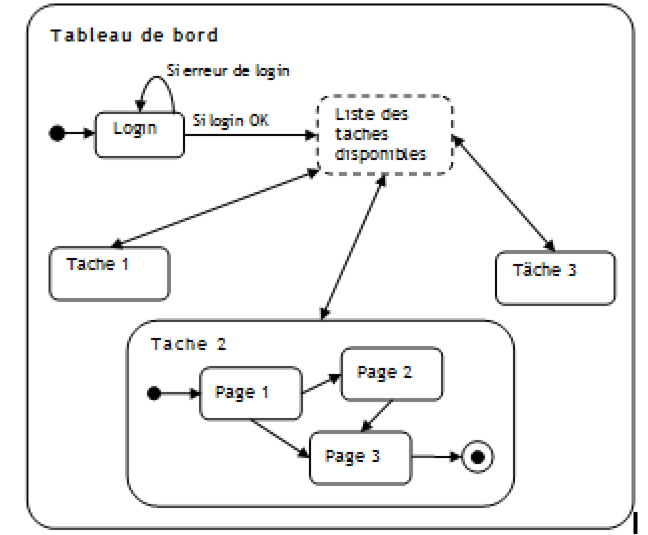


Figure 2. Exemple de modèle StateWebCharts : application de type tableau de bord

Réaliser une proposition de structure hypertexte pour le site Web de la cantine. Cette structure sera décrite sur papier, avec le formalisme StateWebCharts.

Questions complémentaires

Je suis **satisfait** de ma production
☐ Pas du tout ☐ Plutôt non ☐ Plutôt oui ☐ Tout à fait

Je pense que les pages et la structure hypertexte que j'ai produits seraient **utiles** pour la conception de ce site Web
☐ Pas du tout ☐ Plutôt non ☐ Plutôt oui ☐ Tout à fait

Mes productions seraient **lisibles** et **critiquables** par tous les intervenants de la conception de l'interface, du client au développeur
☐ Pas du tout ☐ Plutôt non ☐ Plutôt oui ☐ Tout à fait

Les informations nécessaires pour la spécification de l'interface sont **présentes**
☐ Pas du tout ☐ Plutôt non ☐ Plutôt oui ☐ Tout à fait

Modéliser la structure hypertexte aide à voir l'application dans sa **globalité**
☐ Pas du tout ☐ Plutôt non ☐ Plutôt oui ☐ Tout à fait

Je **représente** la structure hypertexte de l'application lors de sa spécification.
☐ oui ☐ non
Si oui avec quel outil / quelle notation ? ______________________________

Mon avis sur le formalisme StateWebCharts

Les difficultés	Les avantages	Les inconvénients

Il y a des informations que j'aurais **souhaité spécifier** et qui n'apparaissent ni sur les pages, ni sur le modèle SWC ? ☐ oui ☐ non
Si oui lesquelles ?

SCÉNARIO 2

> **Pour le même site Web, d'autres propositions vous sont présentées :**
> - **pour les pages du site**
> - **pour la structure hypertexte (description avec StateWebCharts)**
>
> **Les comparer avec vos propositions et apporter d'éventuelles modifications pour produire une proposition optimale.**

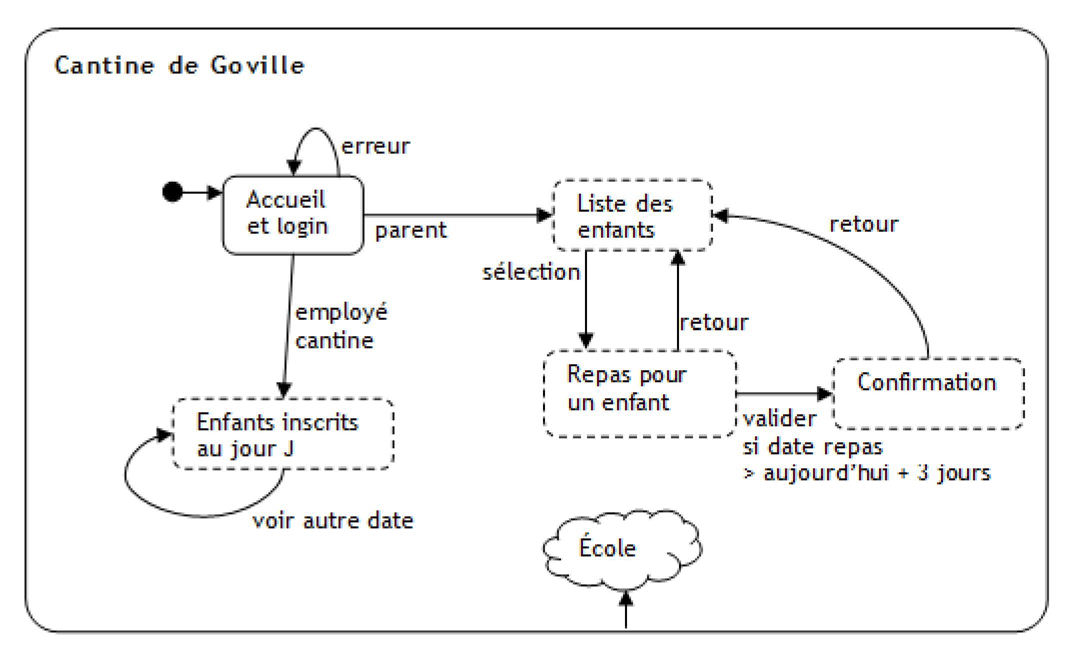


Figure 103. Modèle StateWebCharts étalon du site Web de la cantine de Goville



Bienvenue !

Identifiez-vous avec le nom et le mot de passe
qui vous ont été remis à l'école :

Votre nom

Votre mot de passe

Figure 104. Page étalon du site Web de la cantine de Goville : accueil et formulaire d'identification

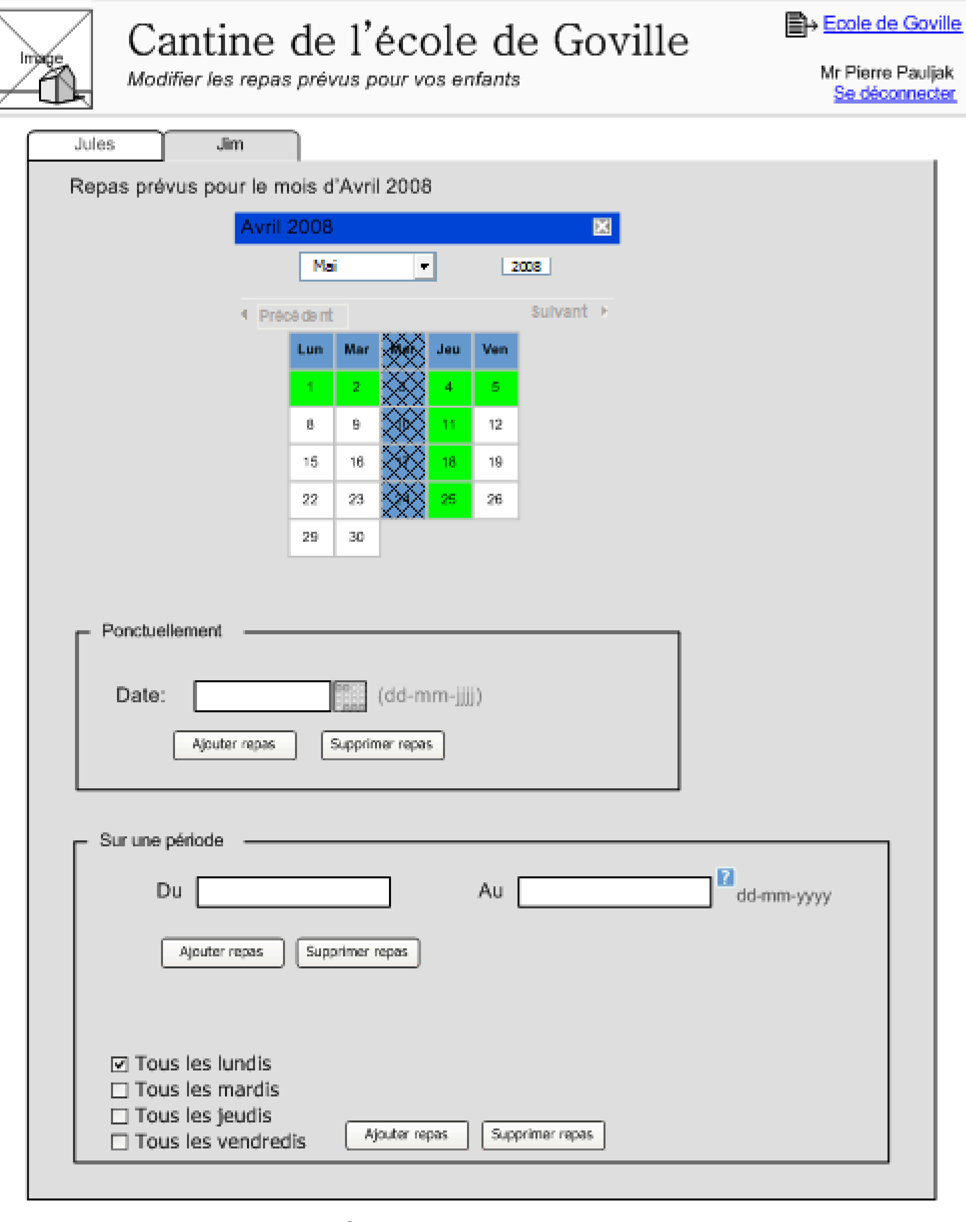


Figure 105. Page étalon du site Web de la cantine de Goville : gestion par un parent des repas de ses enfants

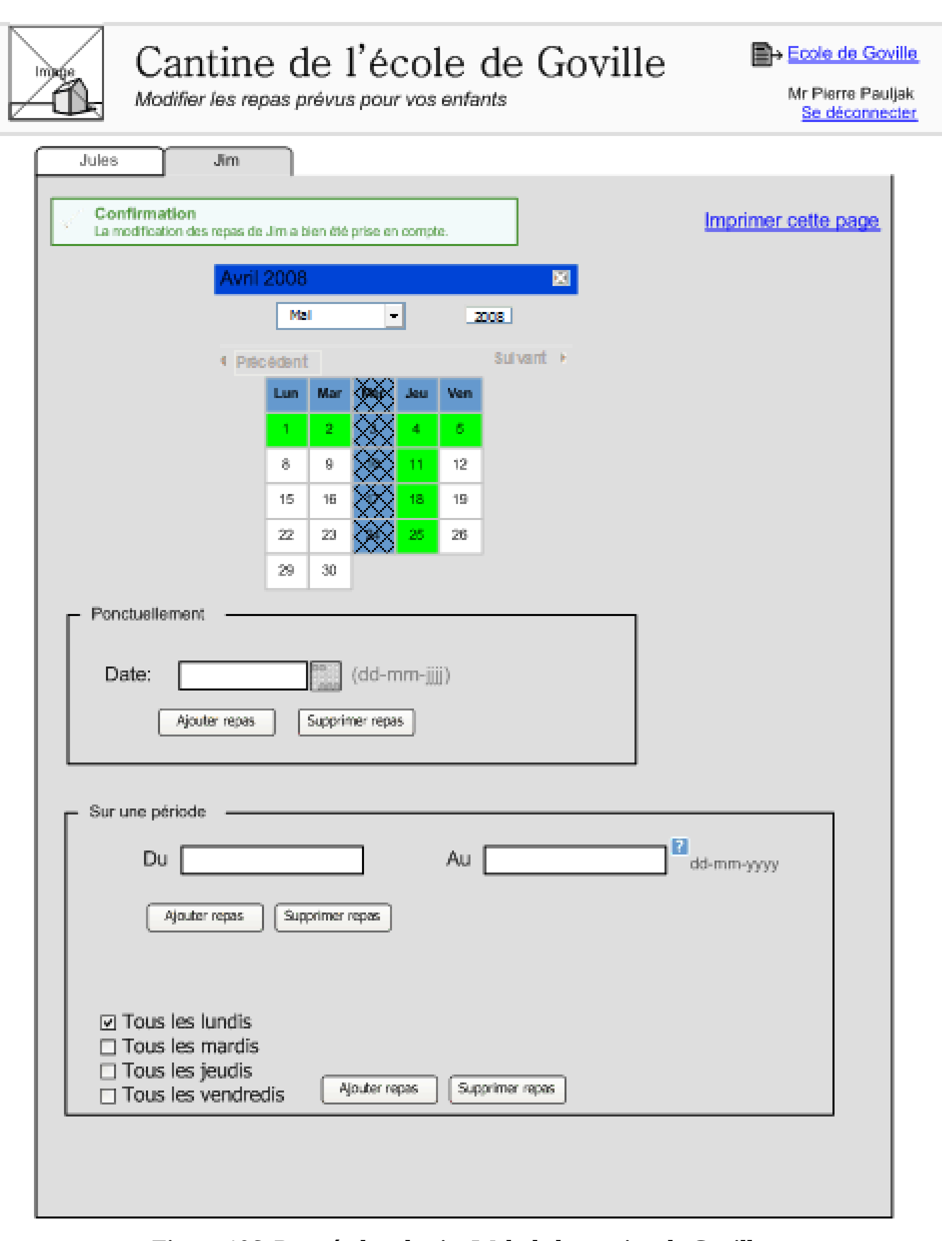


Figure 106. Page étalon du site Web de la cantine de Goville : confirmation de l'ajout ou de la suppression d'un ou de repas

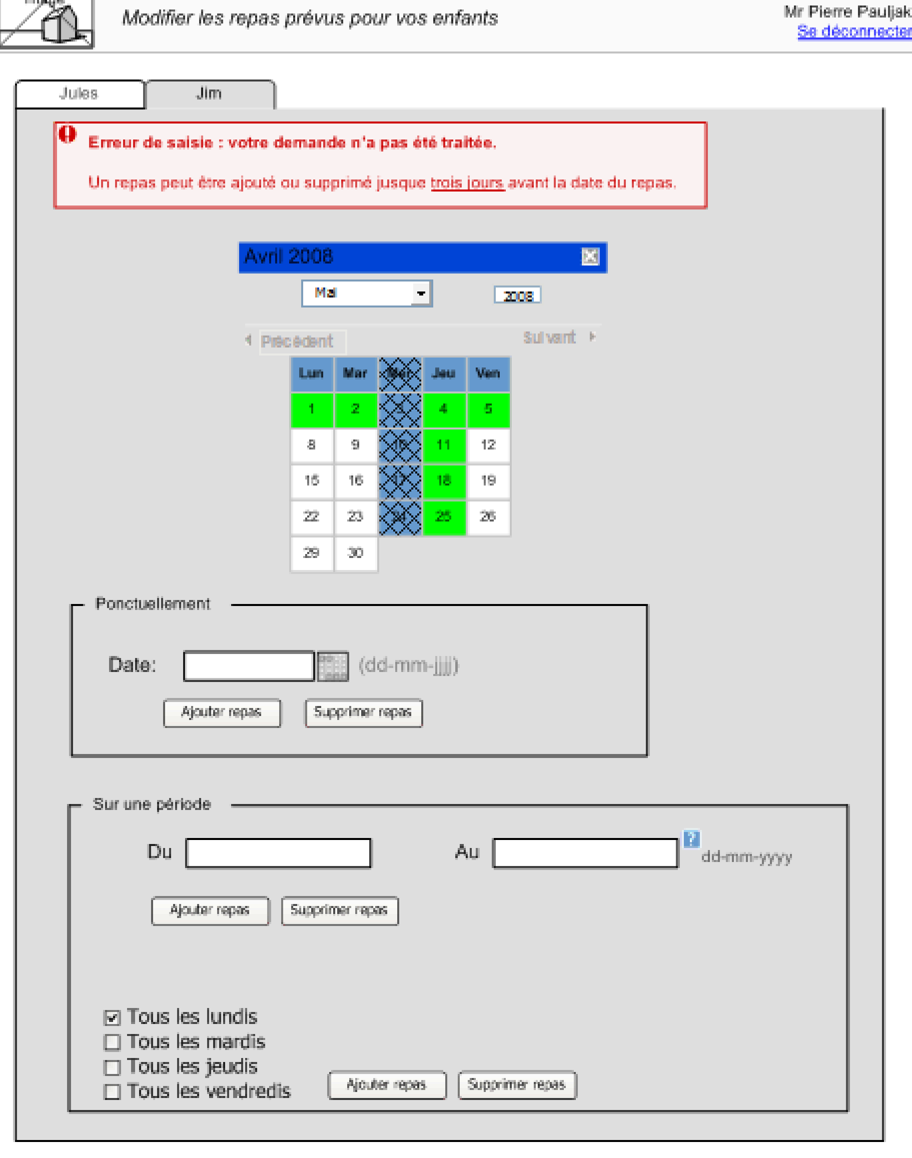


Figure 107. Page étalon du site Web de la cantine de Goville :
signalement d'une erreur survenue lors de l'ajout ou de la suppression d'un ou de repas

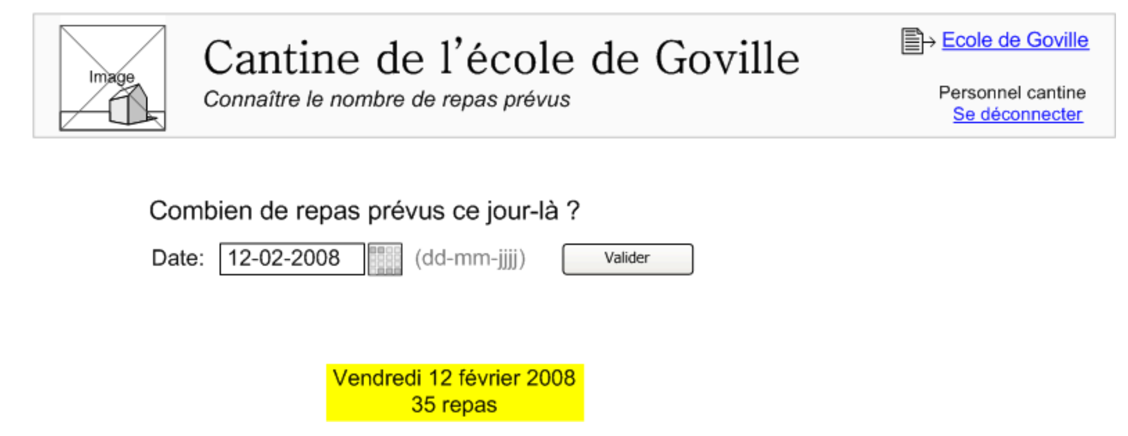


Figure 108. Page étalon du site Web de la cantine de Goville :
affichage, pour le personnel de la cantine, du nombre de repas prévus à une date donnée

SCÉNARIO 3

Sur le même site Web, il faut intégrer la possibilité de payer la cantine.

- *Site web de la cantine de l'école communale de Goville*
- *Accès par identifiant pour un parent pour inscrire ses enfants à la cantine certains jours (l'identifiant est remis en mains propres)*
- *Un repas peut être ajouté ou annulé jusque trois jours avant la date du repas*
- *Les responsables de la cantine peuvent consulter le nombre de repas prévus le jour même ou les jours à venir*
- **Un parent identifié peut régler le montant dû par carte de crédit (les repas à venir pour le mois)**

Proposer un guideline à ajouter au StyleGuide, pour ce module de paiement par carte de crédit, en remplissant le canevas proposé.

Canevas de guideline fonctionnel pour une page

Titre du guideline

Recommandations

Rationnel

Cas d'utilisation (au moins un)

- Doit
  (ce guideline **doit** être utilisé dans les cas suivants)
- Devrait
  (ce guideline **devrait** être utilisé dans les cas suivants)
- Ne devrait pas
  (ce guideline **ne devrait pas** être utilisé dans les cas suivants)
- Ne doit pas
  (ce guideline **ne doit pas** être utilisé dans les cas suivants)

Exemples (au moins un)

- Positifs
  (exemples à **suivre**)

- Négatifs
  (exemples à **ne pas** suivre)

Ressources externes

Prototype
(aperçu de la page telle qu'il est recommandé de la concevoir)

Questions complémentaires

Les éléments suivants me font généralement réaliser qu'un guideline n'apparaît pas dans le StyleGuide :

Proposer un ajout de guideline m'a semblé **facile**
☐ Pas du tout ☐ Plutôt non ☐ Plutôt oui ☐ Tout à fait

Proposer un ajout de guideline m'a semblé **gratifiant**
☐ Pas du tout ☐ Plutôt non ☐ Plutôt oui ☐ Tout à fait

Proposer un ajout de guideline a requis un **temps raisonnable**
☐ Pas du tout ☐ Plutôt non ☐ Plutôt oui ☐ Tout à fait

Je pourrais envisager, dans mon activité quotidienne, de **consacrer** ce temps à proposer un ajout de guideline lorsque l'occasion se présente
☐ Pas du tout ☐ Plutôt non ☐ Plutôt oui ☐ Tout à fait

Je pense que mes connaissances et expériences sont **utiles** pour mes collègues
☐ Pas du tout ☐ Plutôt non ☐ Plutôt oui ☐ Tout à fait

Je me sens **responsable** de l'évolution du contenu du StyleGuide
☐ Pas du tout ☐ Plutôt non ☐ Plutôt oui ☐ Tout à fait

Autres remarques

ENTRETIEN FINAL

Entretien final

Si je pouvais proposer **UNE chose à encourager** dans le StyleGuide...

Si je pouvais proposer **UNE chose à améliorer** dans le StyleGuide...

J'attendais qu'on me pose la question suivante, et elle ne m'a pas été posée

Observations générales
Sur le StyleGuide

Sur le déroulement de l'évaluation

Autres

Je laisse mon **email** pour être tenu au courant des résultats de l'enquête

____

Annexe E. Résultats du pré questionnaire

Le Tableau 11 montre les réponses au pré questionnaire, indexées comme suit :

- Sujet
  - Homme (H) ou Femme (F)
  - Âge (en années)
- Fréquence d’utilisation d’outils
  - Tous les jours (4)
  - Plusieurs fois par semaine (3)
  - Plusieurs fois par mois (2)
  - Plus rarement (1)
  - Jamais (0)
- Statut
- Expérience chez SmalS (en mois)
  - Moins de 6 mois (3)
  - De 6 mois à 1 an (9)
  - De 1 an à 3 ans (24)
  - Plus de 3 ans (36)
- Langues maîtrisées
  - Je ne connais pas cette langue (0)
  - J'ai des notions élémentaires (1)
  - Je peux tenir une conversation (2)
  - Je maîtrise cette langue (3)
- Éléments facilitant l’utilisation du StyleGuide
  - Le sujet en a profité (O) ou pas (N)
- Activités menées avec le StyleGuide
  - Jamais (0)
  - 1 fois (1)
  - 2 ou 3 fois (2,5)
  - 4 fois et plus (4)
- Avis subjectif sur le StyleGuide
  - Pas du tout d'accord (-2)
  - Plutôt pas d'accord (-1)
  - Plutôt d'accord (1)
  - Tout à fait d'accord (2)

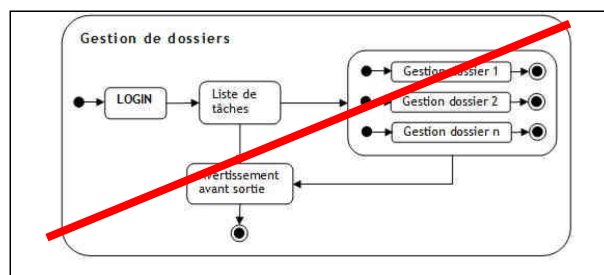
Tableau 11. Résultats quantitatifs du pré questionnaire

	Sujet		Fréquence d'utilisation d'outils						Langues			Eléments facilitant l'utilisation du StyleGuide				Activités menées avec le StyleGuide						Utilisateurs connus du StyleGuide					Avis subjectif										
	Sexe	Age	Editeur de schémas	Editeur web	Navigateur	Bureautique	**Statut**	**Expé. SmalS (en mois)**	Français	Néerlandais	Anglais	A reçu une formation	Connaît le référent conseil	Possède les compétences requises	Intégration aux habitudes de travail	Participation à un projet	Exploration par curiosité	Recherche d'un problème précis	Envoi de commentaire sur le StyleGuide	Personnalisation d'un *wireframe*	Argumentation se référant au StyleGuide	Responsables	Subalternes	Collègues	Client	Personne	Utiliserait même si autre alternative	Conseillerais	Améliore travail	Accélère travail	Permet de gagner en considération	Clarté	Utilité	Accessibilité des *guidelines*	Compréhension	Applicabilité	Comparaison possible entre *guidelines*
1	H	26	3	1	4	4	Analyste	36	3	3	3	N	O	O	O	1,0	1,0	2,5	0,0	4,0	2,5	N	N	N	N	O	-2	-2	-1	-1	-2	1	-1	-1	1	1	1
2	H	24	3	2	4	4	Analyste	3	3	1	2	N	O	O	O	1,0	2,5	4,0	0,0	4,0	4,0	N	N	O	N	N	-1	1	1	-2	1	-1	1	-1	-1	-1	-1
3	F	-	3	1	4	4	Analyste	36	3	2	2	N	O	O	N	1,0	1,0	4,0	0,0	1,0	1,0	N	N	O	N	N	-1	1	1	-1	1	-1	1	1	1	1	-1
4	F	31	1	0	4	4	Chef de projet	36	3	2	2	O	O	O	O	0,0	2,5	2,5	2,5	0,0	0,0	N	O	N	N	N	2	2	2	2	-1	1	1	-1	1	2	1
5	H	24	2	3	4	3	Analyste	24	3	2	2	N	O	O	O	2,5	0,0	2,5	0,0	0,0	0,0	O	N	O	O	N	1	1	1	1	-1	1	1	1	1	1	-1
6	H	24	3	0	4	4	Analyste	9	3	1	2	N	O	O	O	1,0	1,0	0,0	0,0	0,0	4,0	N	N	O	N	N	-1	1	1	-2	1	-1	1	1	1	1	1
7	H	26	2	1	4	4	Analyste	3	3	1	3	N	O	O	O	1,0	1,0	4,0	0,0	1,0	1,0	N	N	O	N	N	-1	2	1	1	1	1	2	1	1	1	1
8	H	25	3	1	4	4	Analyste	24	3	2	3	O	O	O	N	0,0	1,0	0,0	0,0	0,0	0,0	N	N	O	N	N	-2	-1	1	-1	-2	-1	1	1	1	1	1
9	F	49	2	2	4	4	Analyste	36	3	2	2	N	O	O	O	0,0	1,0	1,0	0,0	0,0	0,0	N	N	N	N	O	1	1	1	-1	-1	2	2	1	1	-1	-1
10	H	25	2	1	4	4	Analyste	9	3	2	2	O	O	O	O	0,0	1,0	1,0	0,0	0,0	0,0	O	N	O	N	N	1	2	2	2	-	-	1	1	1	1	-
11	H	25	3	0	4	4	Analyste	24	3	2	2	N	O	O	O	1,0	0,0	2,5	0,0	0,0	1,0	N	N	O	N	N	1	1	1	1	-1	1	1	1	1	1	1
MOYENNE	H	28,6	2,4	1,2	4,0	3,9	Analyste	23,0	3,0	1,8	2,3	N	O	O	O	0,8	1,2	2,3	0,3	1,1	1,4	N	N	O	N	N	-0,4	0,7	0,9	-0,4	-0,3	0,2	1,0	0,3	0,8	0,7	0,1

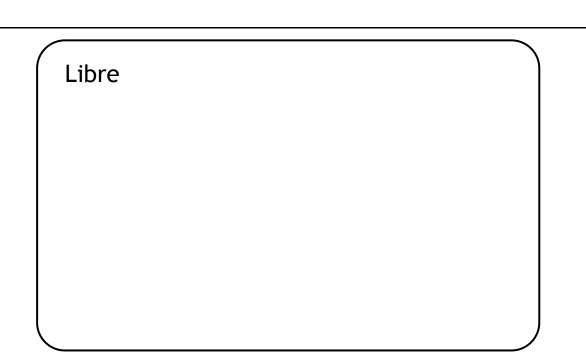
Annexe F. Patrons couverts par eGovIPM

		Base pour un nouveau *wireframe*	Apparaît ou peut être ajouté à un *wireframe*
Patrons d'enchaînement d'écrans	~~Gestion de dossiers~~	~~V~~	~~X~~
	Libre	V	X
	Login	X	V
	Tableau de bord	V	X
	Wizard avec guidage fort	V	X
	Wizard avec guidage souple	V	X
	Wizard avec récapitulatif éditable	V	X
	TOTAL	**5**	**1**
Patrons de page	Accusé de réception	X	V
	Avertissement avant sortie	X	V
	Formulaire de contact	X	V
	Liste des informations à rassembler	X	V
	Liste de tâches	X	V
	Page de formulaire	X	V
	Page de login	X	V
	Page de wizard	X	V
	Récapitulatif	X	V
	TOTAL	**0**	**9**

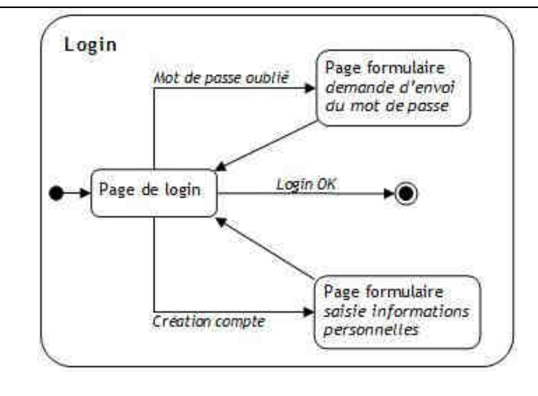
Non pris en compte dans le dévelop-
pement de l'outil eGovIPM
(voir plus bas)

<table>
<tr>
<td>
</td>
<td>Description : Le patron d'enchaînement d'écrans Gestion de dossiers représente une application grâce à laquelle l'utilisateur (généralement un agent administratif) a la possibilité d'agir sur plusieurs dossiers en parallèle, l'action pouvant être simple ou complexe, sur une ou plusieurs pages (modification de données, remplissage de formulaire…)

Réalisation :

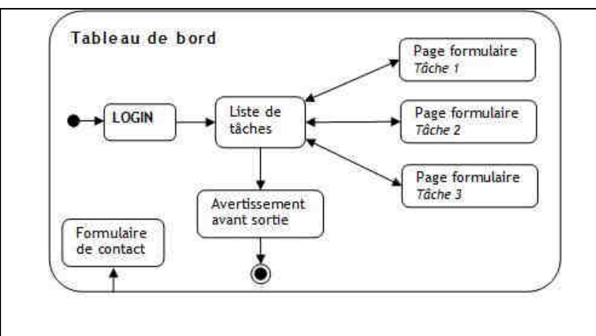
- Restrictions : le patron d'enchaînement d'écrans Gestion de dossiers n'est pas pris en compte par eGovIPM. Les états concurrents ne sont descriptibles dans l'éditeur SWCeditor qu'à hauteur de deux états au maximum : cette restriction ôte tout intérêt à l'implémentation de ce patron dont le but est de démontrer comment un agent administratif peut gérer un grand nombre de dossiers en parallèle.</td>
</tr>
<tr>
<td>
</td>
<td>Description : ce patron vide permet de réaliser un enchaînement d'écrans librement, sans contrainte d'édition, mais en profitant de la fonctionnalité « génération de squelette »

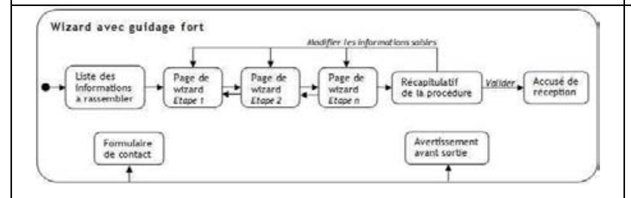
Réalisation :

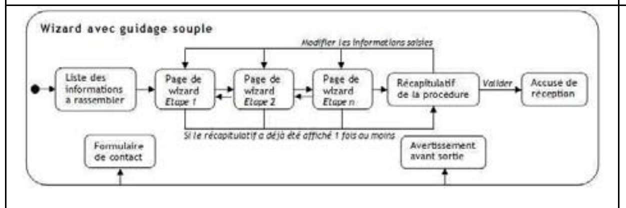
- Note : Le patron d'enchaînement d'écrans Libre n'est attaché à aucune contrainte d'édition.</td>
</tr>
<tr>
<td>
</td>
<td>Description : Ce patron décrit la procédure d'identification d'un visiteur : des informations personnelles peuvent ensuite être affichées. Si l'utilisateur a oublié son mot de passe, une page de formulaire lui permet de saisir une information personnelle preuve de son identité, pour recevoir de nouveau son mot de passe par email. Si l'utilisateur n'a pas de compte, il peut en demander un (si la nature de l'application le permet) via un formulaire.

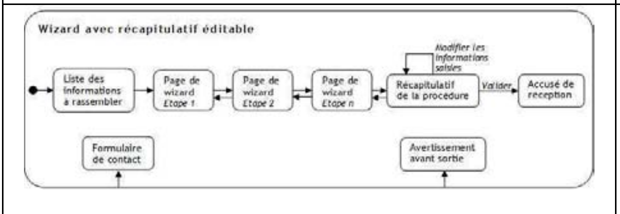
Réalisation :

- Note : Le patron d'enchaînement d'écrans Login est le seul à se comporter dans eGovIPM comme un patron de page. Il ne peut pas servir de base à un wireframe personnalisé ; par contre, il peut apparaître ou être ajouté à un wireframe d'enchaînement d'écrans.</td>
</tr>
</table>

<table>
<tr>
<td>


</td>
<td>
Description : Ce patron Tableau de bord représente un enchaînement d'écrans dans lequel l'utilisateur a à sa disposition un ensemble de tâches, après d'être identifié via un enchaînement d'écrans de Login. Après avoir réalisé une tâche (via une Page de formulaire), l'utilisateur revient à la liste des tâches qu'on lui propose, pour en exécuter une autre ou bien quitter l'application.
Réalisation :
Restrictions : Tel qu'il apparaît dans notre catalogue, ce patron peut faire figurer un enchaînement d'écrans comme tâche à réaliser (ex. un wizard avec guidage souple). Dans cette première version d'eGovIPM, nous considérons, pour simplifier, qu'une tâche peut seulement être une Page de formulaire et non un enchaînement d'écrans.
</td>
</tr>
<tr>
<td>


</td>
<td>
Description : Ce patron Wizard avec guidage fort propose une série d'étapes à suivre pour réaliser un objectif. Une fois que le récapitulatif des informations saisies a été validé, un accusé de réception est affiché. Pour modifier une de ces informations, l'utilisateur est forcé de revenir à l'étape concernée par la modification, et de traverser de nouveau chacune des étapes qui la suivent.
Réalisation :
RAS
</td>
</tr>
<tr>
<td>


</td>
<td>
Description : Même propos dans ce Wizard avec guidage souple que dans le patron Wizard avec guidage fort, mais lors d'une modification, le comportement de l'application est différent. L'utilisateur revient vers l'étape concernée, effectue la modification, puis il est autorisé à rejoindre directement le récapitulatif pour valider ou appliquer d'autres modifications.
Réalisation :
RAS
</td>
</tr>
<tr>
<td>


</td>
<td>
Description : Même propos dans ce Wizard avec guidage avec récapitulatif éditable que dans les patrons Wizard avec guidage fort et Wizard avec guidage souple, mais lors d'une modification, le comportement de l'application est différent. L'utilisateur modifie directement le récapitulatif qui offre des champs éditables pour les informations qui ont été saisies.
Réalisation :
RAS
</td>
</tr>
</table>

Annexe G. Correspondance entre modèle State-WebCharts et page XHTML

SWC	XHTML
État statique E	**Page XHTML**
Id d'un état statique E	Nom de la page XHTML : page_idE.html
Label d'un état statique E	Titre de la fenêtre <head><title>labelE</title></head> Titre principal de la page HTML, apparaît en tête de page <p class="pagetitle">labelE</p> Texte réactif d'un lien vers E sortant de E', apparaît après <p class="smalltitle">List of outgoing links</p> <p class="linklist"> labelE //autres liens </p>
Stéréotype d'un état statique E	Complément d'informations surs un lien vers E sortant d'E' <p class="smalltitle">List of outgoing links</p> <p class="linklist"> labelE stereotypeE //autres liens </p>
Pagetype d'un état statique E	Type de page, apparaît après le titre de la page <p class="pagetype">pagetypeE</p> Si le champ image est vide, détermine l'adresse de l'image à afficher pour le *wire-frame* <p class="smalltitle">*Wireframe*</p>
Description d'un état statique E	Description de la page, apparaît après le type de la page descriptionE
Image d'une page	Si le champ image est vide, le *wireframe* générique apparaît, fichier image dont le nom est déterminé par l'attribut pagetype (cf. Pagetype) Sinon, le *wireframe* qui apparaît est le fichier JPG spécifié dans le champ image <p class="smalltitle">*Wireframe*<p>
Etat composé E	**Enchaînement de pages XHTML**
Id d'un état composé E	Ø
Label d'un état composé E	Ø
Stéréotype d'un état composé E	Ø
Screenflowtype d'un état composé E	Ø
Etat dynamique	Ø
Etat externe	Ø
Etat transitoire	Ø
Etat avec historique	Ø
Transition utilisateur	**Lien hypertexte**
Label d'une transition	Condition de l'activation d'une transition affichée sur l'état source <p class="linklist"> If labelT: labelE //autres liens </p>
Stéréotype d'une transition	Ø
Etat source Esource d'une transition	Page XHTML dans laquelle le lien apparaît : page_idEsource.html
Etat cible Ecible d'une transition	Page XHTML ciblée par le lien hypertexte symbolisé par la transition labelEcible
Transition dynamique	Ø

Annexe H. Grammaire du format XML de description des enchaînements d’écrans

Cette grammaire enrichit la grammaire utilisée par SWCEditor pour l’édition de modèles de navigation StateWebCharts. Les attributs ajoutés pour le contexte d’eGovIPM sont encadrés.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://liihs.irit.fr/swceditor"
targetNamespace="http://liihs.irit.fr/swceditor" elementFormDefault="qualified">

<!--============================== ELEMENTS ==============================-->
<xs:element name="Swc">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="CompositeState" minOccurs="1" maxOccurs="1"/>
			<xs:element ref="Transition" minOccurs="0" maxOccurs="unbounded"/>
		</xs:sequence>
		<xs:attribute name="project" use="optional" type="xs:string"/>
		<xs:attribute name="author" use="optional" type="xs:string"/>
		<xs:attribute name="date" use="optional" type="xs:string"/>
		<xs:attribute name="version" use="optional" type="xs:string"/>
	</xs:complexType>
</xs:element>

<xs:element name="CompositeState">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="GraphicBounds" minOccurs="0" maxOccurs="1"/>
			<!--<xs:element ref="InitialState" minOccurs="0" maxOccurs="1"/>-->
			<xs:choice minOccurs="0" maxOccurs="unbounded">
				<xs:element ref="CompositeState"/>
				<xs:element ref="StaticState"/>
			</xs:choice>
			<xs:element ref="DeepHistory" minOccurs="0" maxOccurs="1"/>
			<xs:element ref="ShallowHistory" minOccurs="0" maxOccurs="1"/>
			<xs:element ref="EndState" minOccurs="0" maxOccurs="1"/>
		</xs:sequence>
		<xs:attribute name="id" use="required" type="xs:string"/>
		<xs:attribute name="label" use="optional" type="xs:string"/>
		<xs:attribute name="file" use="optional" type="xs:string"/>
		<xs:attribute name="initial" use="required" type="xs:boolean"/>
		<xs:attribute name="concurrent" use="required" type="xs:boolean"/>
		<xs:attribute name="entryAction" use="optional" type="xs:string"/>
		<xs:attribute name="doAction" use="optional" type="xs:string"/>
		<xs:attribute name="exitAction" use="optional" type="xs:string"/>
		<xs:attribute name="stereotype" use="optional" type="xs:string"/>
		<xs:attribute name="screenflowtype" use="optional" type="CompositeStateType"/>
	</xs:complexType>
</xs:element>

<xs:element name="StaticState">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="GraphicBounds" minOccurs="0" maxOccurs="1"/>
		</xs:sequence>
		<xs:attribute name="id" use="required" type="xs:string"/>
		<xs:attribute name="label" use="optional" type="xs:string"/>
		<xs:attribute name="type" use="required" type="StaticStateType"/>
		<xs:attribute name="file" use="optional" type="xs:string"/>
		<xs:attribute name="initial" use="required" type="xs:boolean"/>
		<xs:attribute name="entryAction" use="optional" type="xs:string"/>
```

```
		<xs:attribute name="doAction" use="optional" type="xs:string"/>
		<xs:attribute name="exitAction" use="optional" type="xs:string"/>
		<xs:attribute name="stereotype" use="optional" type="xs:string"/>
		<xs:attribute name="description" use="optional" type="xs:string"/>
		<xs:attribute name="image" use="optional" type="xs:string"/>
		<xs:attribute name="pagetype" use="optional" type="StaticStatePageType"/>
	</xs:complexType>
</xs:element>

<xs:element name="DeepHistory">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="GraphicBounds" minOccurs="0" maxOccurs="1"/>
		</xs:sequence>
		<xs:attribute name="id" use="required" type="xs:string"/>
	</xs:complexType>
</xs:element>

<xs:element name="ShallowHistory">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="GraphicBounds" minOccurs="0" maxOccurs="1"/>
		</xs:sequence>
		<xs:attribute name="id" use="required" type="xs:string"/>
	</xs:complexType>
</xs:element>

<xs:element name="EndState">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="GraphicBounds" minOccurs="0" maxOccurs="1"/>
		</xs:sequence>
		<xs:attribute name="id" use="required" type="xs:string"/>
	</xs:complexType>
</xs:element>

<xs:element name="Transition">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="Point" minOccurs="0" maxOccurs="1"/>
		</xs:sequence>
		<xs:attribute name="id" use="required" type="xs:string"/>
		<xs:attribute name="type" use="required" type="TransitionType"/>
		<xs:attribute name="label" use="optional" type="xs:string"/>
		<xs:attribute name="source" use="required" type="xs:string"/>
		<!-- The target must be either a CompositeState OR a State OR an EndState-->
		<xs:attribute name="target" use="required" type="xs:string"/>
		<xs:attribute name="trigger" use="optional" type="xs:string"/>
		<xs:attribute name="guard" use="optional" type="xs:string"/>
		<xs:attribute name="action" use="optional" type="xs:string"/>
		<xs:attribute name="stereotype" use="optional" type="xs:string"/>
	</xs:complexType>
</xs:element>

<xs:element name="GraphicBounds">
	<xs:complexType>
		<xs:attribute name="x" use="optional" type="xs:integer"/>
		<xs:attribute name="y" use="optional" type="xs:integer"/>
		<xs:attribute name="width" use="optional" type="xs:integer"/>
		<xs:attribute name="height" use="optional" type="xs:integer"/>
	</xs:complexType>
</xs:element>

<xs:element name="Point">
	<xs:complexType>
```

```
                <xs:attribute name="x" use="required" type="xs:integer"/>
                <xs:attribute name="y" use="required" type="xs:integer"/>
        </xs:complexType>
</xs:element>

<!--============================== TYPES ==============================-->

<xs:simpleType name="CompositeStateType">
        <xs:restriction base="xs:string">
                <xs:enumeration value="dashboard"/>
                <xs:enumeration value="login"/>
                <xs:enumeration value="strong_wizard"/>
                <xs:enumeration value="supple_wizard"/>
                <xs:enumeration value="editable_overview_wizard"/>
        </xs:restriction>
</xs:simpleType>

<xs:simpleType name="StaticStateType">
        <xs:restriction base="xs:string">
                <xs:enumeration value="static"/>
                <xs:enumeration value="transient"/>
                <xs:enumeration value="dynamic"/>
                <xs:enumeration value="external"/>
        </xs:restriction>
</xs:simpleType>

<xs:simpleType name="StaticStatePageType">
        <xs:restriction base="xs:string">
                <xs:enumeration value="acknowledgement_of_receipt"/>
                <xs:enumeration value="contact_form"/>
                <xs:enumeration value="form_page"/>
                <xs:enumeration value="checklist"/>
                <xs:enumeration value="tasks_set"/>
                <xs:enumeration value="login_page"/>
                <xs:enumeration value="overview"/>
                <xs:enumeration value="warning_before_exit"/>
                <xs:enumeration value="wizard_page"/>
        </xs:restriction>
</xs:simpleType>

<xs:simpleType name="TransitionType">
        <xs:restriction base="xs:string">
                <xs:enumeration value="user"/>
                <xs:enumeration value="system"/>
                <xs:enumeration value="completion"/>
        </xs:restriction>
</xs:simpleType>

</xs:schema>
```

AUTHOR: Florence PONTICO

TITLE: "A design method based on interface patterns for e-Government applications"

SUPERVISED BY: Marco WINCKLER, Regina BERNHAUPT, Philippe PALANQUE

DEFENDED ON: Friday, July 11th of 2008, at IRIT (Toulouse, France)

SUMMARY: The design of e-Government applications has to cope with inherent difficulties of the domain. On one hand, the dematerialisation of administrative procedures involves a multidisciplinary design team which needs customised tools to support their discussions. On the other hand, e-Government applications have to be accessible to the wider audience, which requires a lot of attention on the design of user interfaces. In this context, one should note that e-Government applications show lots of similarities, which reinforces the need for interfaces homogenisation. This work studies and proposes a design method allowing the settlement of homogenisation mechanisms in user-centred design of e-Government applications. The reflexion began with a theoretical study of available means to store and display ergonomic knowledge for interface design. This review of existing methods was completed by an ethnographic study in the field, within an IT firm devoted to the design of e-Government applications. On the basis of these observations, this thesis presents: a) a design method based on interface patterns to guide the design of usable e-Government applications; b) a catalogue of patterns providing solutions (rationally or empirically proven) to recurrent interface design problems; c) a tool supporting the personalisation of screen flows patterns; and d) an evaluation on the effective use of interface patterns in the field.

KEYWORDS: Human Computer Interaction, User-centered Web design, e-Gouvernement, Interface design patterns, Web navigation modelling

ADMINISTRATIVE DISCIPLINE: Computer Science

LABORATORY DETAILS:

IRIT (Institut de Recherche en Informatique de Toulouse)
Université Paul Sabatier
118 Route de Narbonne
31062 Toulouse cedex 9
France

AUTEUR : Florence PONTICO

TITRE : « Une méthode de conception basée sur des patrons d'interface pour les applications d'e-Gouvernement »

SOUS LA DIRECTION DE : Marco WINCKLER, Regina BERNHAUPT, Philippe PALANQUE

SOUTENUE LE : Vendredi 11 juillet 2008, à l'IRIT (Toulouse, France)

RESUME : Le développement d'applications d'e-Gouvernement doit surmonter des difficultés inhérentes aux caractéristiques du domaine. D'une part la dématérialisation des procédures administratives fait intervenir une équipe de conception multidisciplinaire qui a besoin d'outils communs adaptés pour dialoguer. D'autre part, les applications doivent être accessibles au plus grand nombre ce qui nécessite une attention toute particulière à la conception des interfaces. Dans ce contexte, il est important d'indiquer que les applications d'e-Gouvernement présentent beaucoup de similitudes, ce qui renforce le besoin d'homogénéisation des interfaces utilisateurs. Ce travail étudie et propose une méthode de conception qui favorise la mise en place de mécanismes d'homogénéisation des interfaces pour des applications d'e-Gouvernement selon une approche centrée utilisateur. La réflexion a été menée sur la base d'une étude théorique des moyens de mise à disposition des connaissances ergonomiques pour la conception d'interfaces. Cette revue de l'existant a été complétée par une étude ethnographique menée dans une entreprise dédiée à la conception d'applications d'e-Gouvernement. Suite à ces études, cette thèse présente : a) une méthode de conception basée sur des patrons d'interfaces qui guide la conception d'applications d'e-Gouvernement utilisables ; b) un catalogue de patron qui propose des solutions (prouvées rationnellement ou empiriquement) à des problèmes de conception d'interfaces connus et récurrents ; c) un outil guidant la personnalisation des patrons d'interface concernant l'enchaînement d'écrans ; et d) une étude sur l'utilisation effective de ces patrons d'interfaces sur le terrain.

MOTS-CLES : Interaction Homme Machine, Conception Web centrée utilisateur, e-Gouvernement, Patrons de conception d'interface, Modélisation de la navigation Web

DISCIPLINE ADMINISTRATIVE : Informatique

INTITULE ET ADRESSE DU LABORATOIRE :

IRIT (Institut de Recherche en Informatique de Toulouse)
Université Paul Sabatier
118 Route de Narbonne
31062 Toulouse cedex 9
France